LSO/PILOT INTERACTION SIMULATOR

Phillip Gary Stueck

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

LSO/PILOT INTERACTION

SIMULATOR

by

Phillip Gary Stueck

Thesis Advisor: M. H. Redlin

June 1973

Approved for public release; distribution unlimited.

LSO/Pilot Interaction Simulator

by

Phillip Gary Stueck
Lieutenant, United States Navy
B.S., Oregon State University, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1973

ABSTRACT

This thesis investigates the feasibility of simulating the Landing Signal Officer (LSO)/pilot interaction during the approach to a landing on an aircraft carrier. A simulator was created which duplicated the LSO's operational environment through the use of computer-generated visual displays. The LSO and the pilot were placed in this simulated carrier approach environment by 1) displaying a representation of the landing area plus a "meatball" and angle of attack information to the pilot while 2) simultaneously displaying the aircraft's approach to the LSO.

Test results demonstrated the basic feasibility of simulating the LSO/pilot interaction and its application as a research tool in studying LSO models, wave-off techniques and landing techniques.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

b	Span of the airplane (ft)
$\bar{c}$	Length of mean aerodynamic chord (ft)
C_D	Drag/$\frac{1}{2}\rho v_t^2 S$
C_{D_0}	Reference coefficient of drag (landing configuration)
C_{D_α}	Coefficient of drag due to a change in angle of attack
C_L	Lift/$\frac{1}{2}\rho v_t^2 S$
$C_{L_{\delta e}}$	$\partial C_L/\partial \delta e$
C_{L_0}	Reference C_L, landing configuration $\left(\frac{mg \cos \theta_0}{P_{d_0}}\right)$
C_{L_α}	Airplane lift-curve slope $\partial C_L/\partial \alpha$
C_ℓ	Rolling-moment coefficient
C_{ℓ_β}	Rolling-moment coefficient due to side slip
C_{ℓ_p}	Roll damping derivative $\partial C_\ell/\partial p$
$C_{\ell r}$	Roll damping derivative $\partial C_\ell/\partial r$
$C_{\ell_{\delta a}}$	Aileron control effectiveness derivative
$C_{\ell_{\delta r}}$	Rolling-moment coefficient due to rudder deflection
C_m	Pitching-moment coefficient $\left[\frac{m}{\frac{1}{2}\rho v_t^2 S\tau}\right]$
C_{m_u}	Change in pitching-moment coefficient due to velocity
C_{m_α}	Change in pitching-moment coefficient due to angle of attack

$C_{m_{\dot{\alpha}}}$	Pitching-moment coefficient due to the rate of change of angle of attack
C_{m_q}	Pitching-moment coefficient due to roll rate
$C_{m_{\delta e}}$	Pitching moment coefficient due to elevator deflection
C_n	Yawing-moment coefficient
C_{n_β}	Yawing-moment coefficient due to side slip
Cn_p	Yawing cross derivative
Cn_r	Yaw damping derivative
$Cn_{\delta a}$	Aileron yaw derivative
$Cn_{\delta r}$	Rudder control effectiveness derivative
Cy	Side force coefficient
Cy_β	Side force coefficient due to side slip
Cy_p	Side force coefficient due to rolling
Cy_r	Side force coefficient due to yawing
$Cy_{\delta a}$	Side force coefficient due to aileron deflection
$Cy_{\delta r}$	Side force coefficient due to rudder deflection
$I_{xx}, I_{yy}, I_{zz}, I_{xz}, I_{yz}, I_{xy}$	Moments and products of Inertia about the (X_B, Y_B, Z_B) axis (Slug - ft^2)
m	Mass of the aircraft (slugs)
(P,Q,R)	Scalar components of angular velocity vector of the airplane (radians/sec)
(p,q,r)	Perturbation of (P,Q,R)
P_{d_0}	Dynamic pressure based on $u_o, (\frac{1}{2}\rho u_o^2)$
S	Wing area (ft^2)
T	Thrust
u_0	Reference flight speed (ft/sec)

(U,V,W)	Scalar components of v_c(ft/sec)
(u,v,w)	Perturbations of (U,V,W)
v_c	Velocity vector of airplane mass center
v_t	True airspeed
(X',Y',Z')	- Coordinate of airplane mass center relative to fixed axes
(X_{OB},Y_{OB},Z_{OB})	Object axis coordinates
(X_{VP},Y_{VP},Z_{VP})	Viewing axis coordinates
(X_H, Y_H, Z_H, W_H)	Homogeneous coordinate system
α	Angle of attack (w/u_o)
β	Angle of side slip (v/u_o)
$(\delta e, \delta r, \delta a)$	Angles of elevator, rudder, and aileron deflection
δ_T	Increment of thrust
θ_o	Reference angle of climb
(ξ,η,ζ)	Perturbations of $(\delta a,\delta e,\delta r)$
ρ	Air density (slugs/ft^3)
(Ψ, θ, Φ)	Euler angles, yaw, pitch, roll (radians)
(ψ, θ, ϕ)	Perturbations of (Ψ, θ, Φ)

I. INTRODUCTION

This thesis analyzes the feasibility of creating a simulation of the Landing Signal Officer (LSO)/pilot interaction during an approach to a landing on an aircraft carrier.

Many aspects of the carrier landing system [Fig. 1] have been extensively studied. The pilot-aircraft interface has been investigated with respect to auto-pilot and Automatic Carrier Landing (ACL) systems. The flight dynamics of the aircraft [Ref. 2] and the dynamics associated with the carrier have also been well documented. Aircraft disturbances resulting from wind or stack gas generated burble have already been identified and reproduced. Human describing functions have replaced the pilot in some simulators.

The one area lacking quantitative analysis was the LSO/pilot interface shown as a path parallel to the pilot's input cues in Figure 1. The first systematic study of the Landing Signal Officer was completed by Gail J. Borden in 1969 [Ref. 5,6]. This study stated that one-sixth of all carrier accidents reported to the Naval Safety Center during a five-year period had the LSO listed as either a contributing or causal factor. A more thorough analysis of the LSO's role in the carrier landing system was suggested.

Mr. Borden described many problem areas such as the difficulty LSO's encounter in judging pilot/aircraft approach performance which were identified by means of data received

from opinion surveys. He suggested that, especially in the area of the LSO's preceptual ability, objective validation of the opinion data was necessary. The Naval Air Development Center (NADC) made an analytical attempt to mathematically simulate the behavioral characteristics of the LSO-pilot combination [Ref. 7]. They continued with this program without the benefit of a suitable data base.

Mindful of its potential usefulness, a simulation was created and subjectively evaluated to see if the LSO/pilot interaction environment could be reasonably reproduced. The physical presence of the pilot and the LSO would avoid the problem of trying to mathematically model them. The focus of attention was primarily on displaying to the LSO a picture of the approaching aircraft with enough realism for him to detect changes in lineup, air speed, angle of attack, and position with respect to the glide slope.

Three already published techniques were combined in the construction of the interaction simulation. The aircraft's equations of motion derived by Etkin [Ref. 2] were solved on an analog computer. Techniques in computer processing of three-dimensional structures developed by Desens [Ref. 3] were used to generate the displays shown to the LSO and pilot. The general procedures used by Kahrs [Ref. 4] to display a carrier landing picture to the pilot were used to construct the pilot side of the simulation.

The interactions of the final combined system are shown in Figure 2. The Pilot, with the visual aid of a computer

generated display [Fig. 3] consisting of a runway, angle of attack indicator, and a "meatball", maneuvered the aircraft during the approach to a carrier landing. Concurrently, the LSO was presented with a visual display which depicted the aircraft's approach as seen from the LSO's platform [Fig. 4]. An analog computer solved the aircraft's equations of motion while a digital computer handled the spacial transformations needed for the graphical display [Fig. 2]. The LSO's input to the pilot was aural.

Prior to incorporation into the simulation, the LSO's visual presentation was tested to see if a realistic representation of the aircraft's approach could be displayed. An "O.K." pass was matched for realism with "O.K." passes observed and photographed from the aircraft carrier "Enterprise". A test program was then set up without the pilot in the loop. Each pass was fixed in that the parameters (lineup, angle of attack, glide slope, or air speed) were constant. This preliminary program was then tested for validity by having several subjects judge various approaches to determine parameter sensitivity and real world discrepancies.

The computers used in this simulation were located in Spanagel Hall on the campus of the Naval Postgraduate School, Monterey, California; however, the techniques used were applicable to any comparable computer systems. The computers used were: one analog (CI 5000); one 32K digital (XDS 9300); and two graphics systems (AGT 10). [Ref. 1].

II. ANALOG AND DIGITAL COMPUTER PROGRAMS

A. ANALOG

Reference 2 contains the equations of motion using stability axes, small disturbance theory, and non-dimensionalization. The longitudinal and lateral equations of motion were reduced to a form useful in this simulation. These equations and the assumptions made are presented in Appendix A, and the scaled version of these equations as solved by the CI-5000 analog computer is shown in Appendix B.

Inputs to the analog computer were the outputs of a stick and throttle attached to a seat placed in front of the pilot's graphics displays. The outputs of the analog program were the inertial coordinates of the aircraft (X', Y', Z'), the changes in Euler angle rotation (θ, ϕ, ψ) and changes in the Z' component of velocity (w). The inertial position of the aircraft was then used to locate the aircraft's position with respect to the LSO's position. The rotation angles were used to properly rotate the aircraft in the LSO's display. The velocity component extracted was used to define the airspeed of the aircraft.

B. AXIS MANIPULATION

Using the seven outputs of the analog computer, the digital computer prepares the data sent to the graphics computer for display [Fig. 6]. The critical factor in

processing the inputs for graphical display was the rate at which the picture was updated, i.e. loop time. Taking a movie projector as a norm, a goal of not more than 18 updates per second was set. Therefore, time-saving devices were incorporated whenever possible. Each display will now be discussed separately.

A thesis by R. B. Desens [Ref. 3] showed a method for computer processing a three-dimensional structure for display. A thesis by J. H. Kahrs [Ref. 4] applied Desens' work to displaying the picture seen by the pilot [Fig. 3].

Figure 5 shows an object axis (aircraft axis), an inertial axis (pilot's eye glide slope intersection with carrier's centerline), and a viewing plane axis (the LSO's eye). The following set of matrices were used to transfer a set of points described in the object axis to a set of points in the viewing plane axis. The matrices will also give the view a three-dimensional perspective and control its magnification (scale).

$$[X_H, Y_H, Z_H, W_H] =$$

$$[X_{OB}, Y_{OB}, Z_{OB}, 1] \begin{bmatrix} \text{Object} \\ \text{Axis} \\ \text{Rota-} \\ \text{tion} \end{bmatrix} \begin{bmatrix} \text{Object} \\ \text{Axis} \\ \text{Trans-} \\ \text{lation} \end{bmatrix} \begin{bmatrix} \text{View-} \\ \text{ing} \\ \text{Plane} \\ \text{Axis} \\ \text{Trans.} \end{bmatrix} \begin{bmatrix} \text{View-} \\ \text{ing} \\ \text{Plane} \\ \text{Axis} \\ \text{Rota-} \\ \text{tion} \end{bmatrix} \begin{bmatrix} \text{Perspec-} \\ \text{tive} \\ \text{Offset} \\ \text{Scale} \end{bmatrix}$$

To obtain the display coordinates, division of the homogeneous transformed coordinates must be made by the W_H coordinate. Thus:

$$Y_{DISPLAY} = Y_H/W_H$$

$$Z_{DISPLAY} = Z_H/W_H$$

Where

1. Object Axis Rotation is

$$\begin{bmatrix} A_1 & A_2 & A_3 & 0 \\ B_1 & B_2 & B_3 & 0 \\ C_1 & C_2 & C_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$A_1 = \cos\beta \cos\alpha$

$A_2 = \cos\beta \sin\alpha$

$A_3 = -\sin\beta$

$B_1 = -\cos\gamma \sin\alpha + \sin\gamma \sin\beta \cos\alpha$

$B_2 = \cos\alpha \cos\gamma + \sin\gamma \sin\beta \sin\alpha$

$B_3 = \sin\gamma \cos\beta$

$C_1 = \sin\gamma \sin\alpha + \cos\gamma \sin\beta \cos\alpha$

$C_2 = -\sin\gamma \cos\alpha + \cos\gamma \sin\beta \sin\alpha$

$C_3 = \cos\beta \cos\gamma$

2. Object Axis Translation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ X_{OB} & Y_{OB} & Z_{OB} & 1 \end{bmatrix}$$

3. Viewing Plane Translation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -X_{VP} & -Y_{VP} & -Z_{VP} & 1 \end{bmatrix}$$

4. Viewing Plane Rotation

Same as object axis rotation with

$A_1 = \cos\alpha \cos\beta$

$A_2 = -\sin\alpha \cos\gamma + \sin\gamma \cos\alpha \sin\beta$

$A_3 = \sin\alpha \sin\gamma + \cos\gamma \sin\beta \cos\alpha$

$B_1 = \cos\beta \sin\alpha$

$B_2 = \cos\gamma \cos\alpha + \sin\gamma \sin\beta \sin\alpha$

$B_3 = -\sin\gamma \cos\alpha + \cos\gamma \sin\beta \sin\alpha$

$C_1 = -\sin\beta$

$C_2 = \sin\gamma \cos\beta$

$C_3 = \cos\gamma \cos\beta$

5. Perspective-Offset-Scale

$$\begin{bmatrix} 1 & -Y_o/F & -Z_o/F & -S/F \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & Y_o & Z_o & S \end{bmatrix}$$

Using offset (Y_o, Z_o) it is possible to enlarge any section of the picture. This feature is not used and therefore

$$Y_o = 0$$

$$Z_o = 0$$

The scale factor (S) used for magnification was set at 0.5 (full scale). The focal length (F), used for proper perspective was set for the distance between the viewer and the surface of the CRT scope. The field of view is equal to

$$\tan^{-1}(S/F)$$

The translation and rotation matrices were applied to the problem of displaying the aircraft on the LSO's scope as follows:

a. The object axis was the axis system containing the lines necessary to draw the aircraft. See Fig. 7 for the X_B, Y_B, Z_B axis. See Appendix F for the proper method of describing the outline of the aircraft on data cards.

b. These points were rotated to their new position using the object axis rotation matrix with:

$$\alpha = -\psi$$

$$\beta = \alpha_o + \theta_o + \theta$$

$$\gamma = \phi$$

Since the system was designed to analyze the LSO's function in the loop, it was assumed that the pilot will only make small corrections to his initial configuration and therefore small angle approximations were used to cut down on loop time.

c. The object axis was then translated along the vector OA shown in Fig. 7.

d. The eye of the LSO (viewing plane) was then translated to the LSO platform. See Fig. 7.

e. The viewing plane was then rotated so the LSO is permitted to view the approach from any angle (α) [Fig. 5]. ALPHA equal to zero corresponds to looking directly aft.

Therefore, in the viewing plane rotation:

$$\beta = 0$$

$$\gamma = 0$$

$$\alpha = \text{YVPLSO}$$

This angle of rotation (YVPLSO) is fixed for each approach.

The view of the carrier deck for the pilot's display is similarly generated where:

a. The axis describing the carrier deck is the object axis (X_c, Y_c, Z_c) see Fig. 7.

b. The object axis was not rotated. Therefore, $\alpha = 0$, $\beta = 0$, $\gamma = 0$.

c. The object axis translation was the vector AO. (See Fig. 7)

d. The viewing plane was not translated. Therefore, $X_{VP} = Y_{VP} = Z_{VP} = 0$.

e. The viewing plane is rotated where:

$\alpha = - \psi$ $\qquad$ $\beta = \delta$ $\qquad$ $\gamma = - \phi$

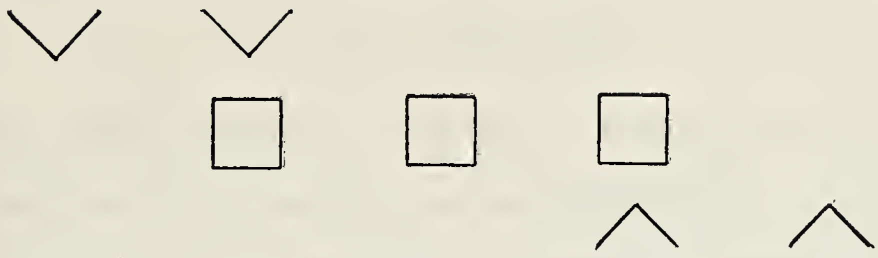
Any line that extends behind the viewer's eye is chopped at its intersection with the plane of the CRT screen. Lines that extend outside the square on the pilot's screen [Fig. 3] or extend off the scope on the LSO's screen are processed through a software window chop [Ref. 3].

Once the lines to be displayed were processed through the matrices, special software subroutines packed the display buffers of the graphics computer and gave the signal to update the display. Figure 6 shows the various steps the digital computer performs in converting the outputs of the analog into pictures displayed to the pilot or LSO.

C. ANGLE OF ATTACK INDICATOR

The angle of attack indicator and the "meatball" [Fig. 3] were displayed as in J. H. Kahr's thesis [Ref. 5]. The five

states of the angle of attack indexer are shown in Fig. 6.



1. 2. 3. 4. 5.

State	Meaning	Range of AOA
1	Slow	Greater than 2.0°
2	Little slow	1.0° to 2.0°
3	Optimum	1.0° to -1.0°
4	Little fast	-1.0° to -2.0°
5	Fast	Less than 2.0°

Figure 8. Angle of Attack Indexer States

D. MEATBALL

The meatball was visible within 1.5 degrees of the optimum glide slope. The meatball presentation [Fig. 3] was located in the middle of the screen for an easier pilot scan. Only the center square moved on the screen. The rectangular shapes on either side of the meatball represented datum lights and did not move.

III. DISCUSSION

The primary obstacle in modeling the system was the time required to regenerate an updated display. Several time-saving features were used to keep the regeneration time less than .065 seconds.

1. A fast subroutine to pack the display buffers was used (RTOI).

2. Small angle approximations were used to draw the aircraft.

3. The number of lines used to draw the aircraft was kept to a minimum.

4. The flight dynamics of the aircraft was solved on an analog computer.

Hidden-line removal [Ref. 3] would have been an aid for drawing a realistic picture of the aircraft; however, its incorporation would have increased the loop time by several milliseconds.

Two expensive hardware update packages for the graphics computers would have diminished the loop time considerably so that other options could be included in the program. The hardware package to window the drawing and the hardware to three-dimensionally rotate an axis system would have been helpful since the software equivalence of these items comprised a major portion of the loop time.

Several potentially enhancing additions were not included in this feasibility study. An audible noise generator linked to the pilot's throttle would aid the LSO in determining the pilot's response to his instructions. A visual representation of the aircraft's exhaust would be helpful for the same reason. Isolating the pilot and LSO stations would decrease distractions.

Once an approach is completed, the LSO critiques the pass and his judgment can be checked against graphs of the actual glide slope or lineup which are displaced on the scope. This feature enables one to determine the ability of the LSO to judge correctly the actual approach.

IV. RESULTS AND CONCLUSIONS

The LSO display was created and tested to see if deviations from the optimum glide-path could be detected. Each approach was run on a constant glide-path which could be above, below, right, or left of optimum. The aircraft's angle of attack could also be changed for each pass.

This preliminary model was tested using five subjects who were not LSO's. Each subject was shown several optimum approaches and several approaches with non-optimum conditions. The student was then given approaches in which he had to recognize the non-optimal condition.

These tests showed that the subjects could judge lineup within 20 feet of center line. The difference between a ramp strike and a bolter was easily recognized. Even angle of attack changes of five degrees or more were noticed. Although the subject was unable to discern deviations from optimal for every approach, his relative difficulty with three parameters (lineup, air speed, glide-slope) was the same as the results of a survey questionnaire given to LSO's which ranked these parameters from least to most difficult to recognize [Ref. 5]. Deviations in altitude were harder to identify than for an actual approach.

The large threshhold of distinction for parameter changes and the lack of ability to determine altitude was

attributed to the lack of training of the student and the lack of a dynamic aircraft.

The analog solution of the flight dynamics of an A-7E was used to give the aircraft the needed motion. The ability of the analog computer to accurately describe a specific aircraft was not germane; however, the model was checked for proper correlation with the A-7E. Correlation of long and short term modes and the ratio of an aileron step input to the change in velocity in the Z' direction are shown below.

	ACTUAL	ANALOG MODEL
Long term	65.8 sec	65.1 sec
Short term	4.85 sec	4.85 sec
$\delta e/DW$	3.17	3.0

The interaction among the pilot, the chair, and the presentation on the pilot's scope was tested by the J. H. Kahr's [Ref. 4]. Kahr's concluded that "sufficient visual cues are provided to enable consistent landings by experienced pilots."

The simulation presented in this thesis demonstrated the feasibility of using a hybrid computer linked to graphic output to present an environment where the interaction of the LSO in the carrier approach landing system could be studied. This program could be used as a basic tool to study the LSO/pilot interaction; however, improved hardware capabilities would enable more extensive studies.

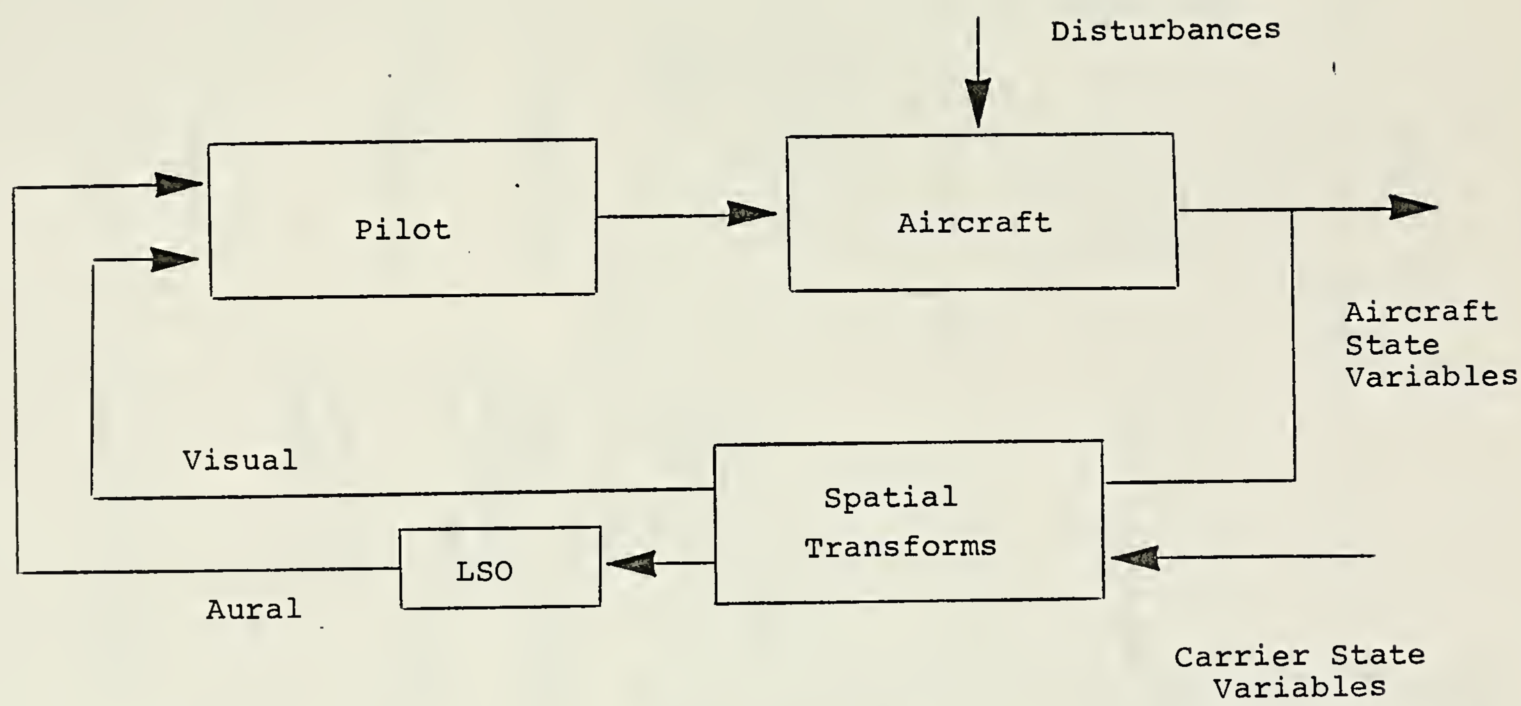


Figure 1. Carrier Landing System Model.

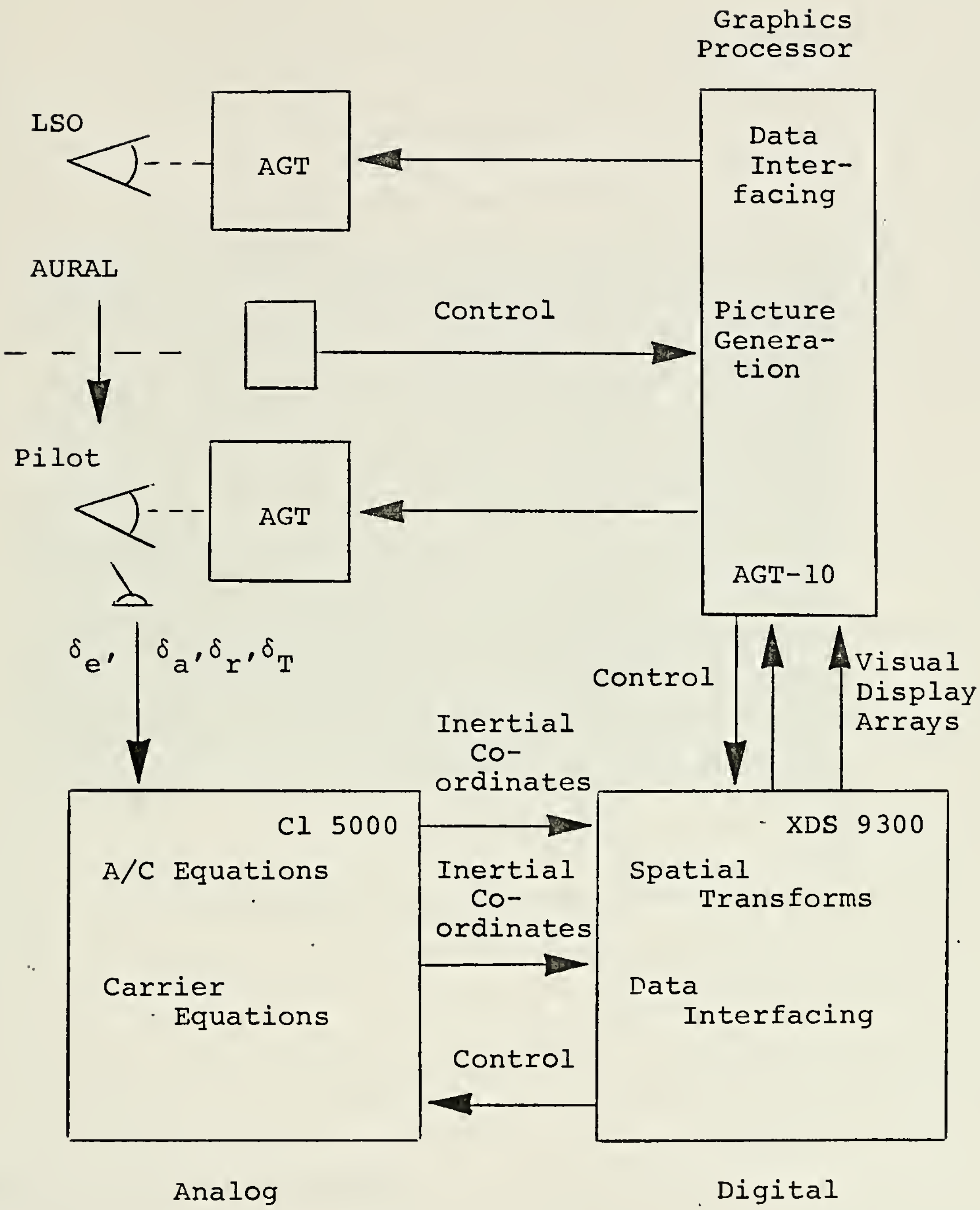


Figure 2. Allocation of Computer Functions.

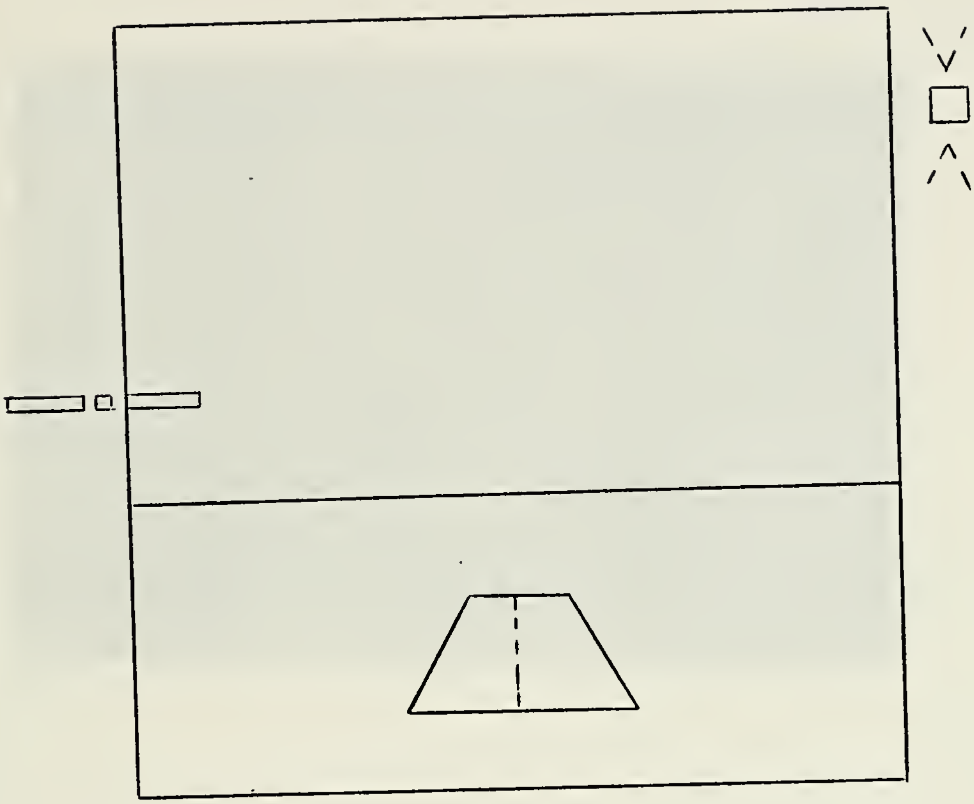


Figure 3. Pilot Display.

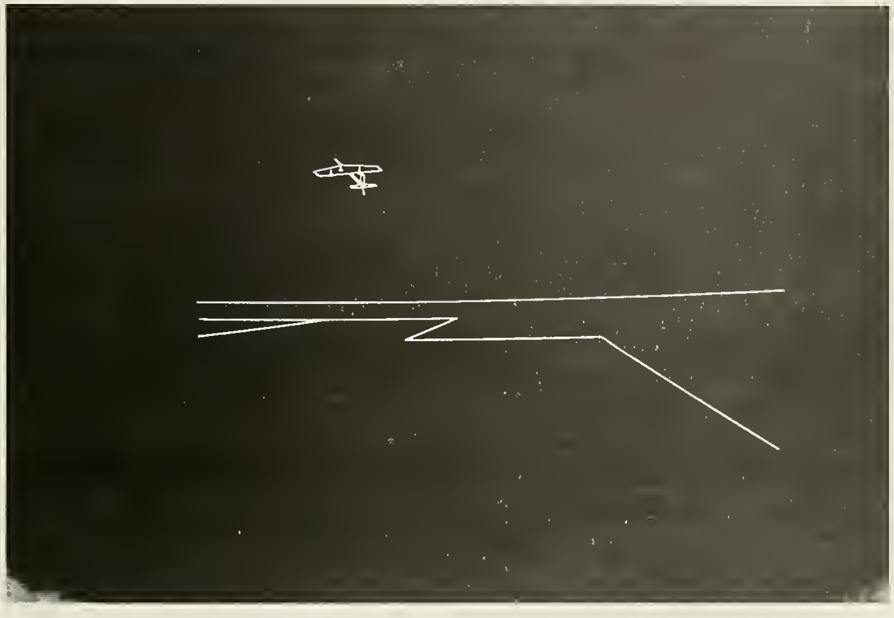


Figure 4. LSO Display.

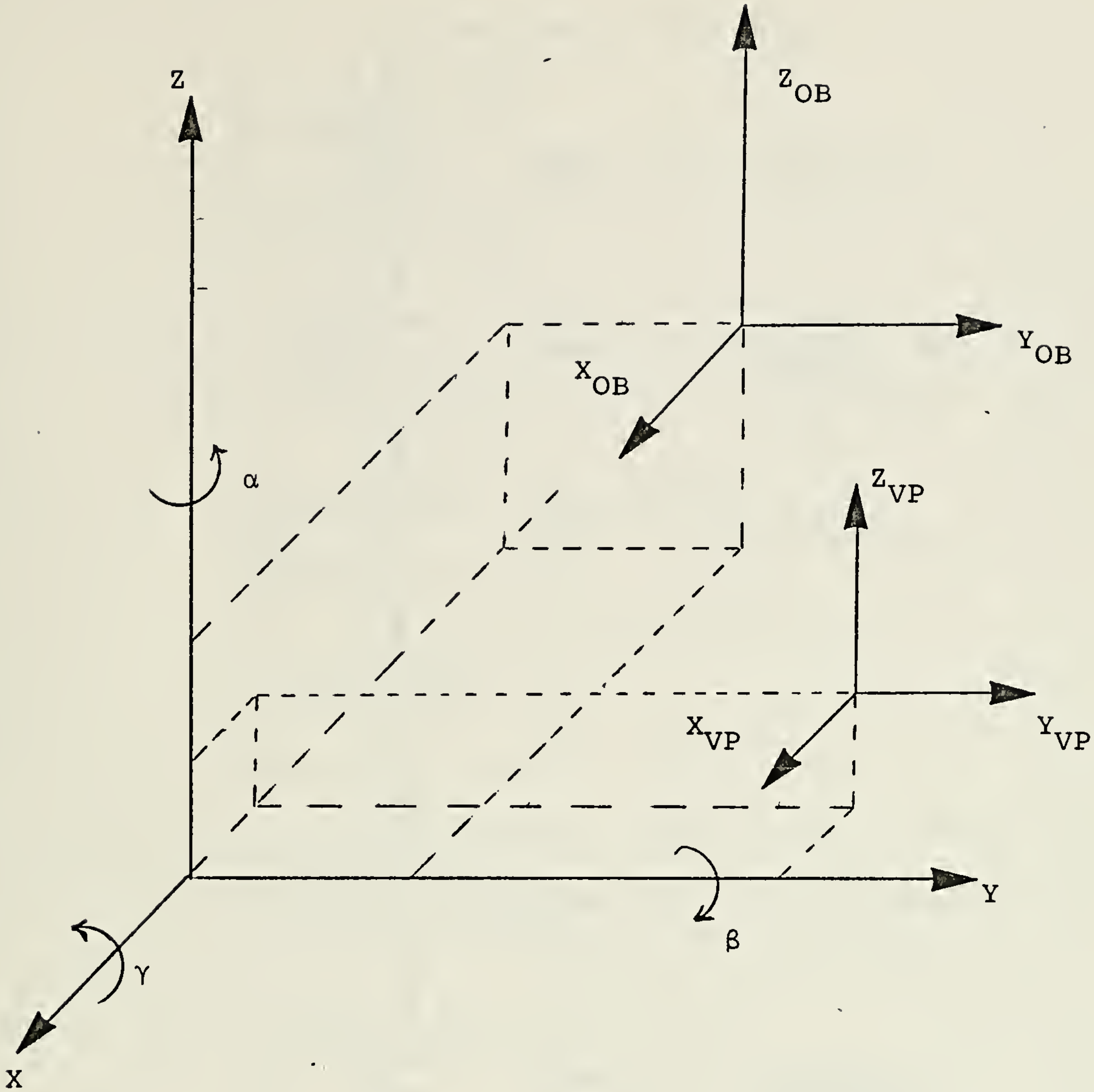


Figure 5. Coordinate Systems.

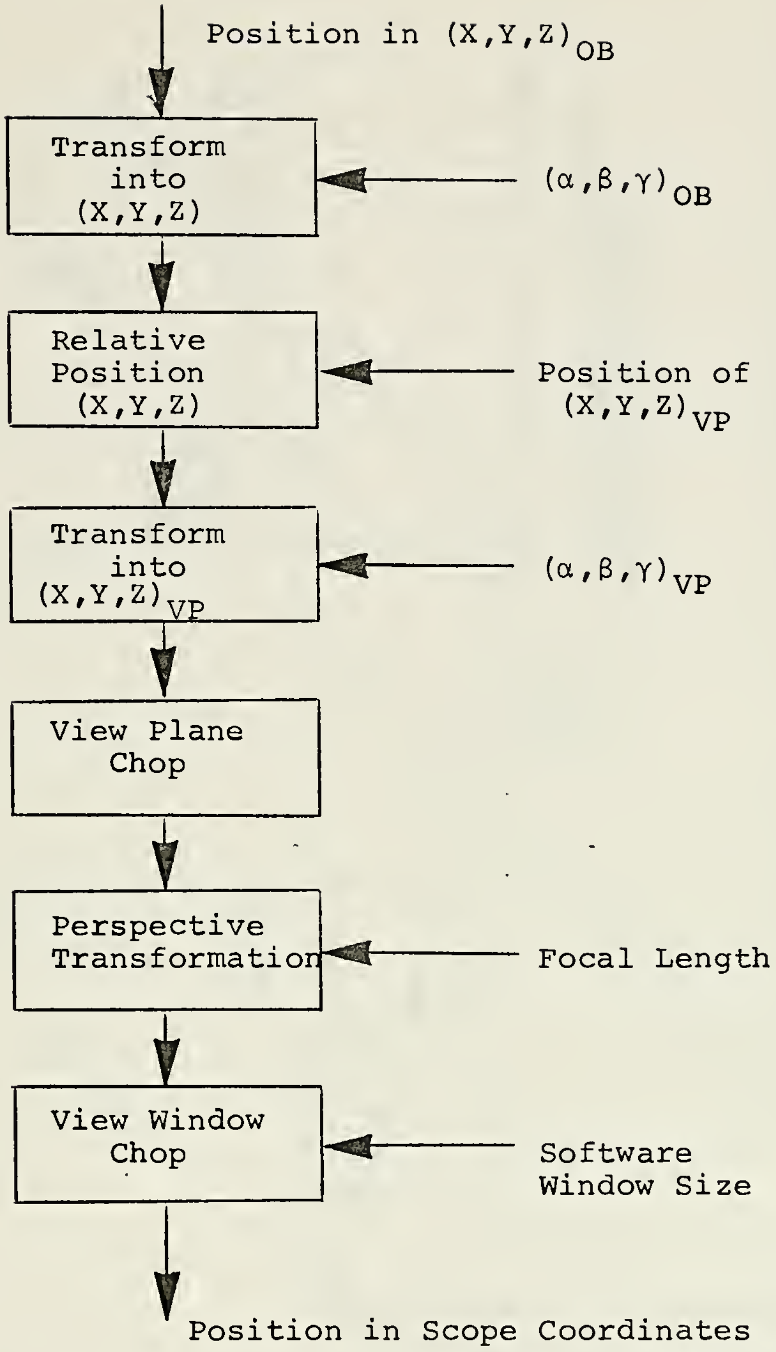


Figure 6. Flow Diagram for Spatial Transformations.

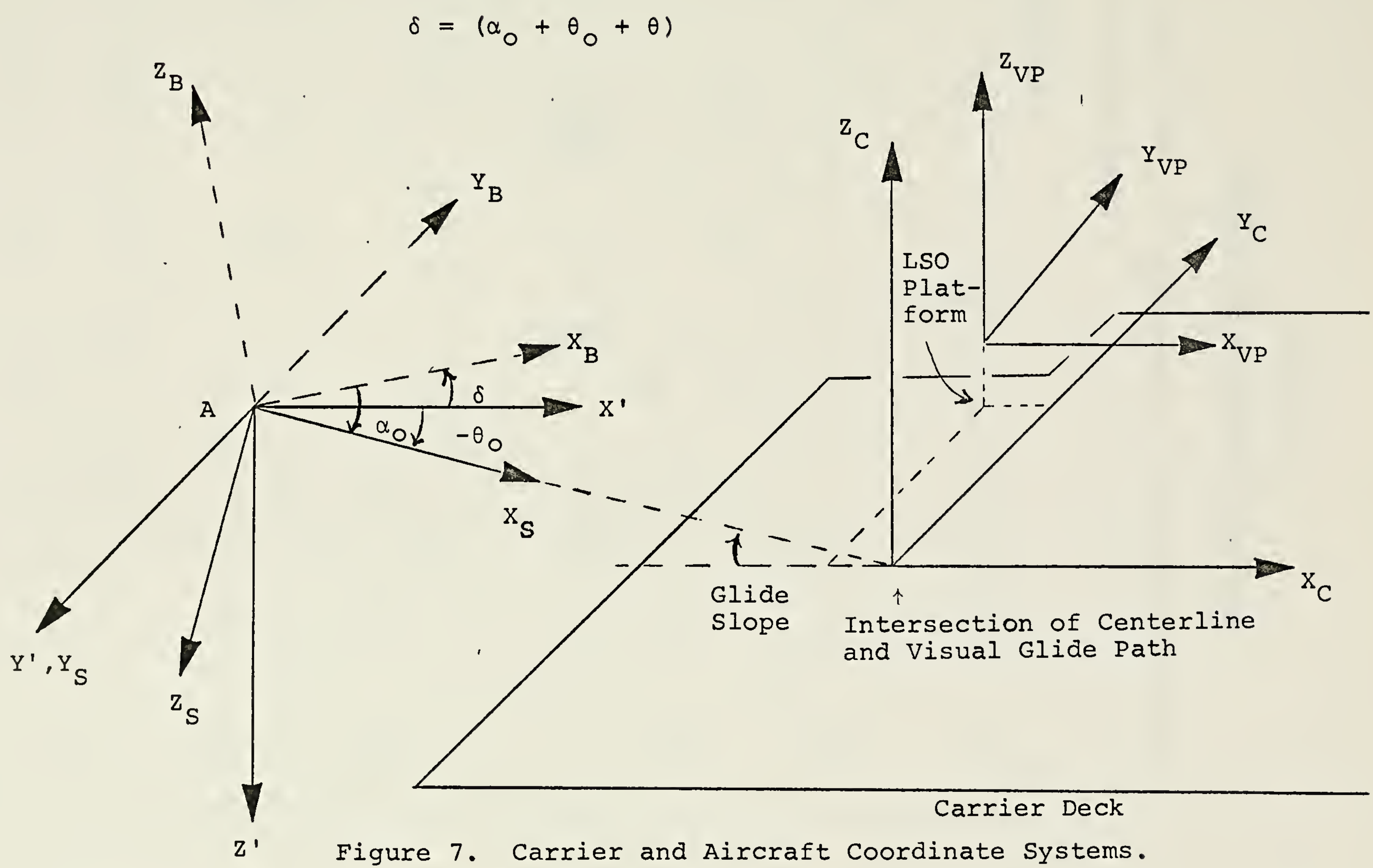


Figure 7. Carrier and Aircraft Coordinate Systems.

APPENDIX A

SUMMARY OF EQUATIONS OF MOTION

Etkin's [Ref. 2] longitudinal and lateral equations are reprinted below:

Longitudinal

$$(2\mu D - 2C_{L_o} \tan\theta_o - C_{x_u})\hat{u} - C_{x\alpha}\alpha + C_{L_o}\theta = 0 \qquad (1)$$

$$(2C_{L_o} + C_{z_u})\hat{u} + (2\mu D - C_{z_{\dot\alpha}} - C_{z_\alpha})\alpha$$
$$-[(2\mu + C_{z_q})D - C_{L_o}\tan\theta_o]\theta - C_{z_\eta}\eta = 0 \qquad (2)$$

$$-C_{m_u}\hat{u} - (C_{m_{\dot\alpha}} D + C_{m_\alpha})\alpha + (i_B D^2 - C_{m_q} D)\theta$$
$$-(C_{m_{\dot\eta}} D + C_{m_\eta})\eta = 0 \qquad (3)$$

where $\mu = \dfrac{m}{\rho S \ell}$ $\quad t^* = \dfrac{\ell}{u_o}$ $\quad \ell = \dfrac{\bar{c}}{2}$

Lateral

$$(2\mu D - C_{y_\beta})\beta - C_{y_p}\hat{p} + (2\mu - C_{y_r})\hat{r} - C_{L_o}\phi$$
$$+ C_{y_\zeta}\zeta = 0 \qquad (4)$$

$$-C_{\ell_\beta}\beta + (i_A D - C_{\ell_p})\hat{p} - (i_E D + C_{\ell_r})\hat{r}$$
$$-(C_{\ell_{\xi}} D + C_{\ell_\xi})\xi - C_{\ell_\zeta}\zeta = 0 \qquad (5)$$

$$-C_{n_\beta}\beta - (i_E D + C_{n_p})\hat{p} + (i_C D - C_{n_r})\hat{r}$$
$$-C_{n_\xi}\xi - (C_{n_\zeta} D + C_{n_\zeta})\zeta = 0 \qquad (6)$$

The following substitutions and assumptions were used to reduce these equations to a more usable form.

In General

$$D = t^* \frac{d}{dt} \qquad \hat{u} = \frac{u}{u_o} \qquad \beta = \frac{v}{u_o}$$

$$U = u + u_o \qquad P_{d_o} = \frac{1}{2} \rho u_o$$

Longitudinal

$$i_b = \frac{I_{yy}(\gamma)}{\rho S \bar{c}^3} \qquad \mu = \frac{2m}{\rho S \bar{c}} \qquad t^* = \frac{\bar{c}}{2u_o}$$

Equation 1

$$C_{x_u} = \frac{(\partial T/\partial u)_o}{P_{d_o} S} - 2(C_{D_o} + C_{L_o} \tan \theta_o) - M \frac{\partial C_D}{\partial M}$$

$$C_{x_\alpha} = C_{L_o} - C_{0_\alpha}$$

where

$$\frac{(\partial T/\partial u)_o}{P_{d_0} S} = 0 \quad \text{and} \quad M \frac{\partial C_D}{\partial M} = 0$$

Equation 2

$$C_{z_u} = - M \frac{\partial C_L}{\partial M} = 0$$

$$C_{z_\alpha} = -(C_{L_\alpha} + C_{D_o})$$

$$C_{z_{\dot{\alpha}}} = 0 \qquad C_{z_q} = 0$$

$$C_{z_\eta} = 0$$

Equation 3

$$C_{m_{\dot{\eta}}} = 0 \qquad C_{m_\eta} = -a_e V_H = C_{m_{\delta e}}$$

Lateral

$$\hat{p} = \frac{pb}{2u_o} \qquad \hat{q} = \frac{q\bar{c}}{2u_o} \qquad \hat{r} = \frac{rb}{2u_o}$$

$$i_a = \frac{I_{xx}(\gamma)}{\rho\ Sb^3} \qquad i_c = \frac{I_{zz}(\gamma)}{\rho\ sB^3}$$

$$i_c = \frac{I_{xz}(\gamma)}{\rho\ Sb^3} \qquad \mu = \frac{2m}{\rho Sb} \qquad t^* = \frac{b}{2u_o}$$

Equation 4

$$C_{y_\zeta}\ \zeta = C_{y_{\delta r}}\ \delta r$$

Equation 5

$$C_{\ell_\xi} = C_{\ell_{\delta a}} \qquad C_{\ell_{\dot{\xi}}} = 0$$

$$C_{\ell_\zeta} = C_{\ell_{\delta r}}$$

Equation 6

$$C_{n_\xi} = C_{n_{\delta a}} \qquad C_{n_{\dot{\zeta}}} = 0 \qquad C_{n_\rho} = C_{n_{\delta r}}$$

As a result of the above substitutions and assumptions, the following equations resulted and were solved by an analog computer.

$$\dot{p} = \frac{P_{d_0} S b}{I_{xx}} \left[\frac{C_{\ell_\beta}}{u_o} v + \frac{C_{\ell_p} b}{2u_o} p + \frac{C_{\ell_r} b}{2u_o} r + C_{\ell_{\delta a}} \delta a + C_{\ell_{\delta r}} \delta r \right] + \frac{I_{xz}}{I_{xx}} \dot{r} \tag{7}$$

$$\dot{r} = \frac{P_{d_0} S b}{I_{zz}} \left[\frac{C_{n_\beta}}{u_o} v + \frac{C_{n_p} b}{2u_o} p + \frac{C_{n_r} b}{2u_o} r + C_{n_{\delta a}} \delta a + C_{r_{\delta r}} \delta r \right] + \frac{I_{xz}}{I_{zz}} \dot{p} \tag{8}$$

$$\dot{v} = \frac{P_{d_0} S}{m} \left[\frac{C_{y_\beta}}{u_o} v + \frac{C_{y_p} b}{2u_o} p + \frac{C_{y_r} b}{2u_o} r + C_{y_{\delta a}} \delta a + C_{y_{\delta r}} \delta r + C_{L_0} \phi \right] - u_o r \tag{9}$$

$$\dot{u} = \frac{P_{d_0} S}{m} \left[\frac{-2C_{D_0}}{u_o} u + (C_{L_0} - C_{D_\alpha}) \frac{\omega}{u_o} - C_{L_0} \theta \right] + \frac{1}{m} \delta_T \tag{10}$$

$$\dot{\omega} = \frac{P_{d_0} S}{m} \left[\frac{-2C_{L_0}}{u_o} u - \frac{(C_{L_\alpha} + C_{D_0})}{u_o} \omega - C_{L_0} \theta \tan \theta_o \right] + u_o q \tag{11}$$

$$\dot{q} = \frac{P_{d_0} S \bar{c}}{I_{yy}} \left[\frac{C_{m_u}}{u_o} u + \frac{C_{m_\alpha}}{u_o} \omega + \frac{C_{m_{\dot{\alpha}}} \bar{c}}{2u_o^2} \dot{\omega} + \frac{C_{m_q} \bar{c}}{2u_o} q + C_{m_{\delta e}} \delta e \right]$$

The remaining equations are reprinted from p. 122 Ref. 2.

$$\theta = q \tag{13}$$

$$\phi = p + r \tan \theta_o \tag{14}$$

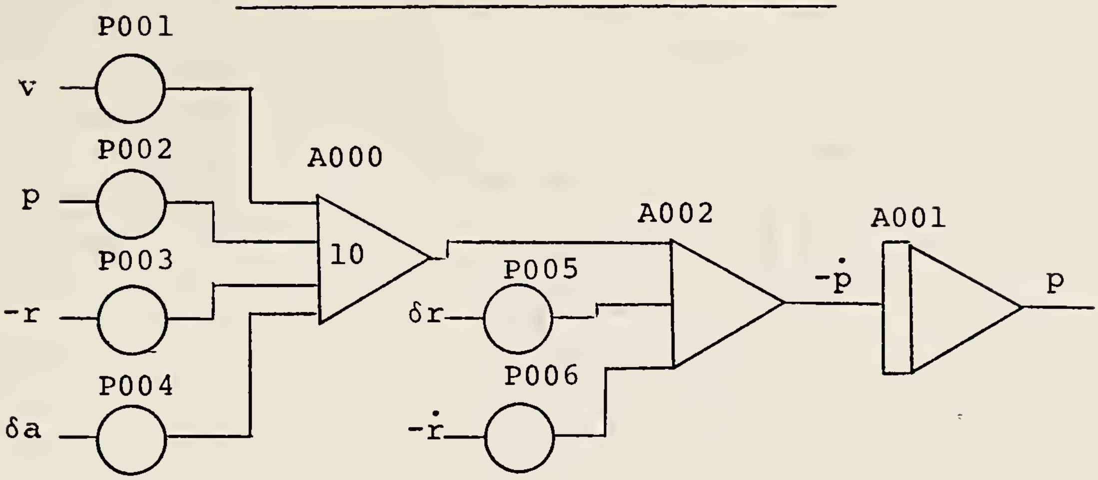
$$\mathring{\psi} = r \sec \theta_o \tag{15}$$

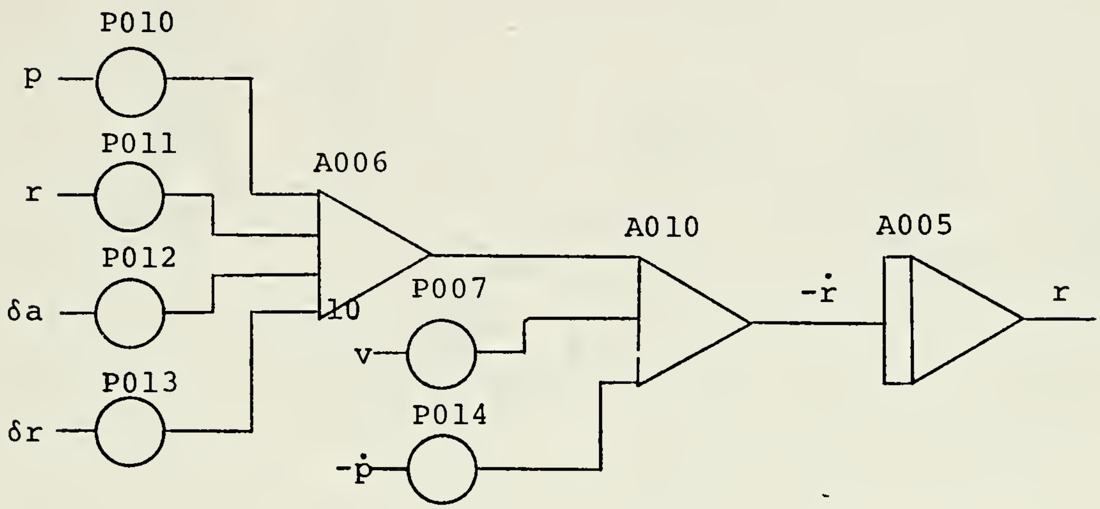
$$\mathring{x} = u_o \cos \theta_o + u \cos \theta_o - u_o \theta \sin \theta_o + \theta_o w \tag{16}$$

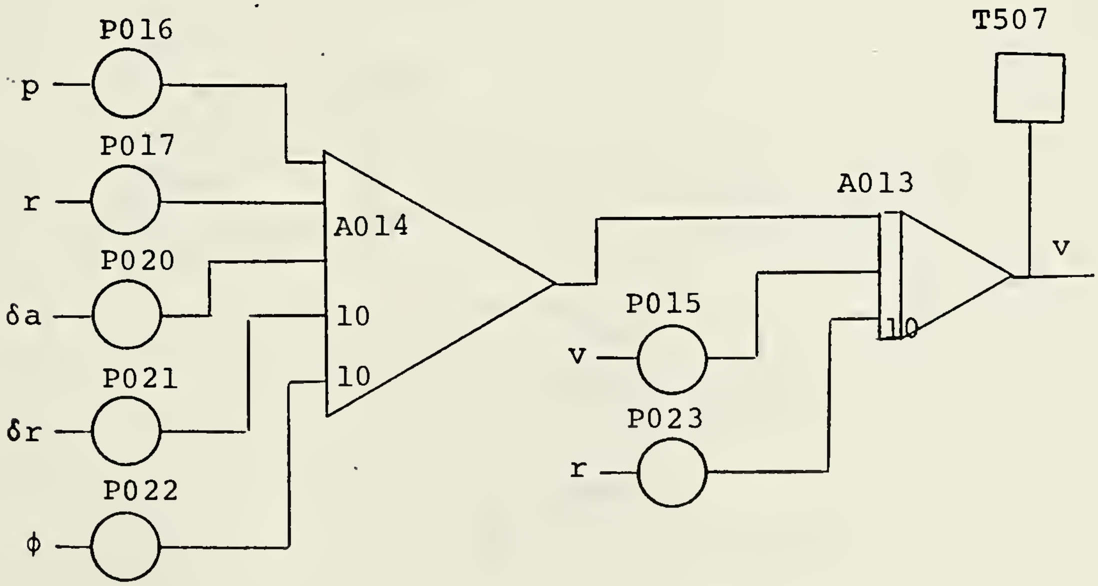
$$\mathring{y}' = u_o \psi \cos \theta_o + v \tag{17}$$

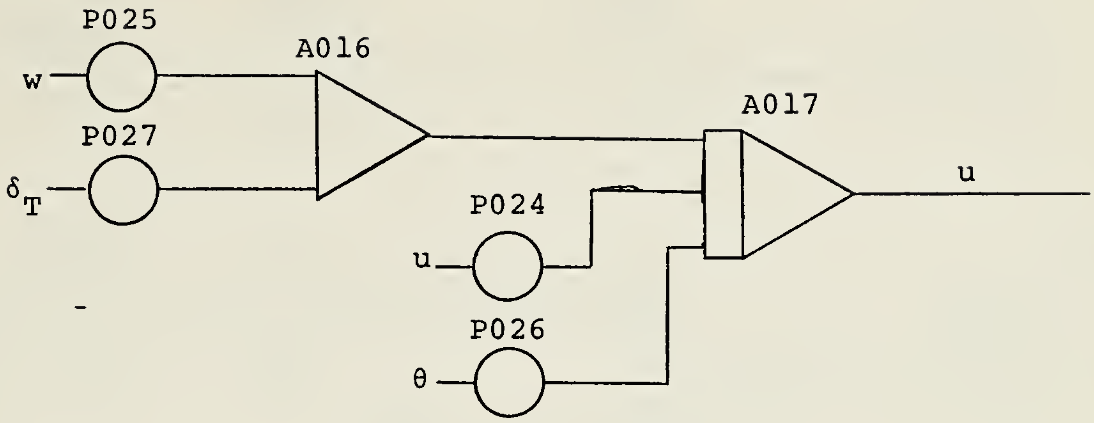
$$\mathring{z}' = - u_o \sin \theta_o - \sin \theta_o u - u_o \theta \cos \theta_o + w \cos \theta_o \tag{18}$$

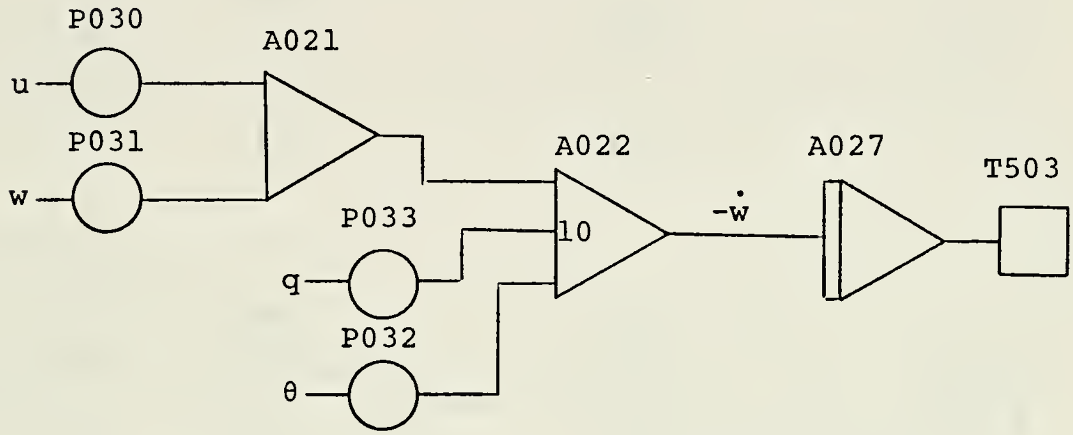
Equations 7-18 were scaled and solved on the Cl-5000 analog computer as shown in Appendix B.

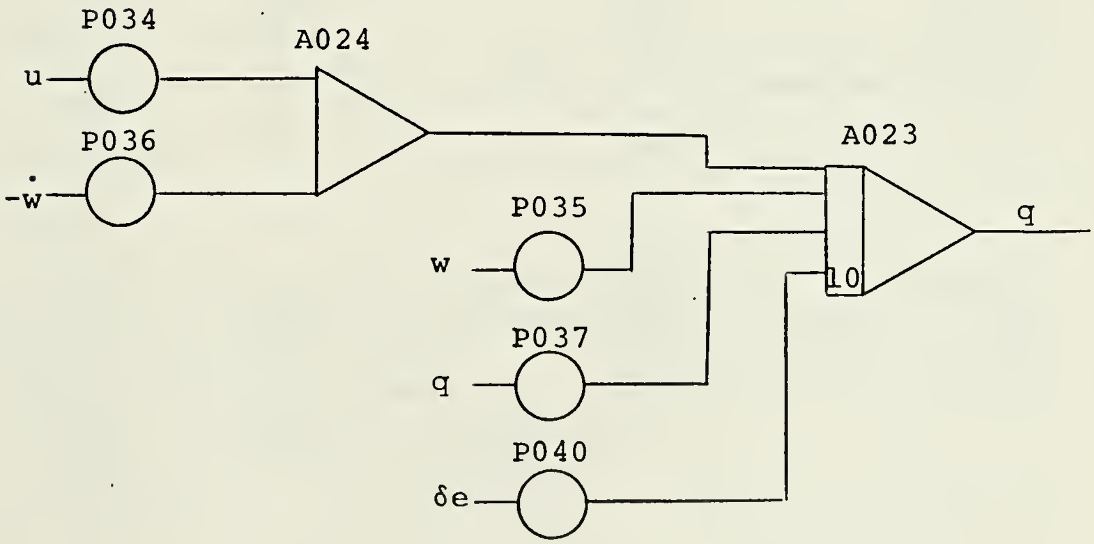

P001
v
P002
p
A000
P003
-r
10
P004
δa
A002
P005
δr
P006
-ṙ
A001
-ṗ
p

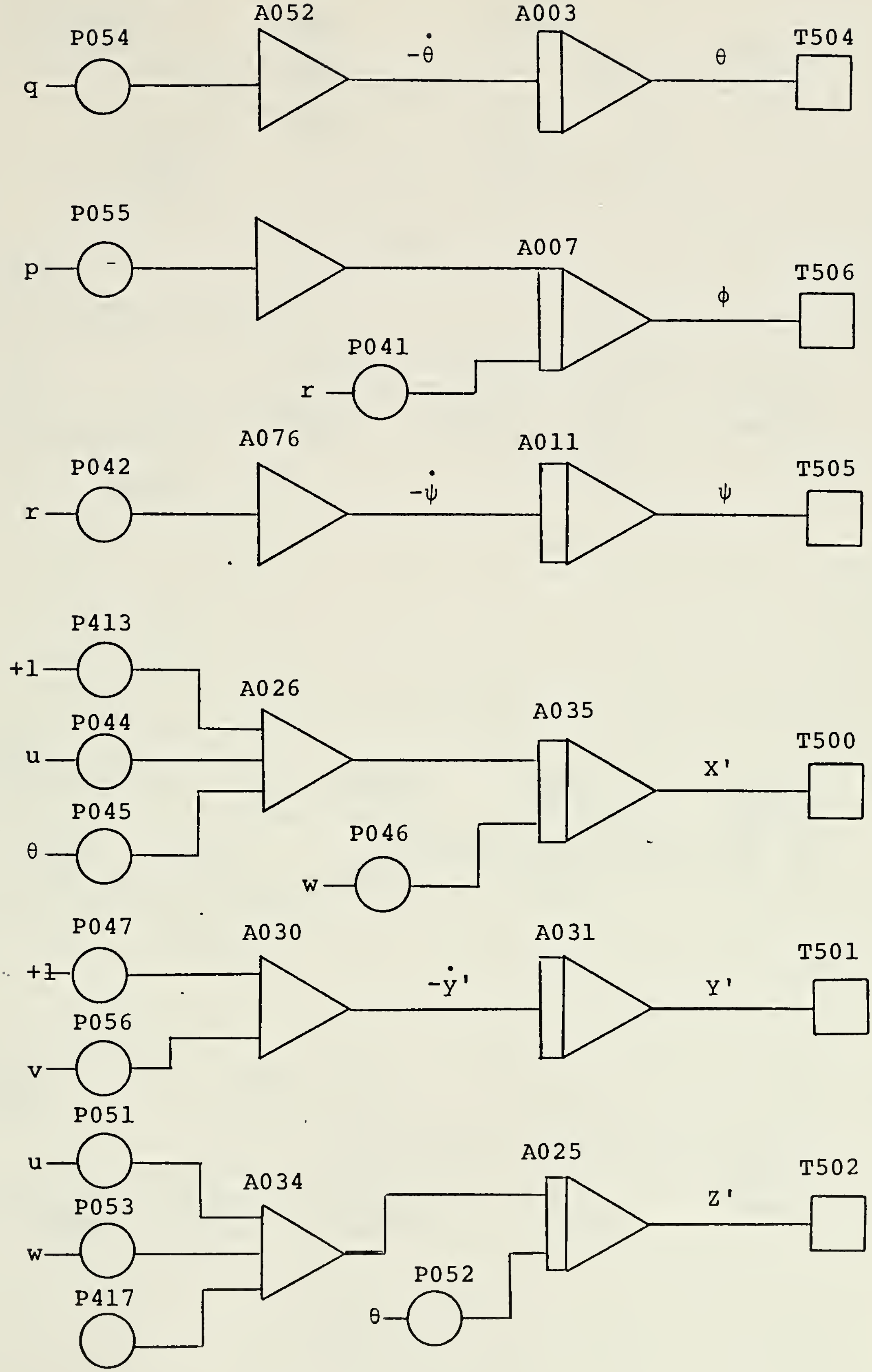

P010
p
P011
r
A006
P012
δa
10
P013
δr
A010
P007
v
P014
-ṗ
A005
-ṙ
r

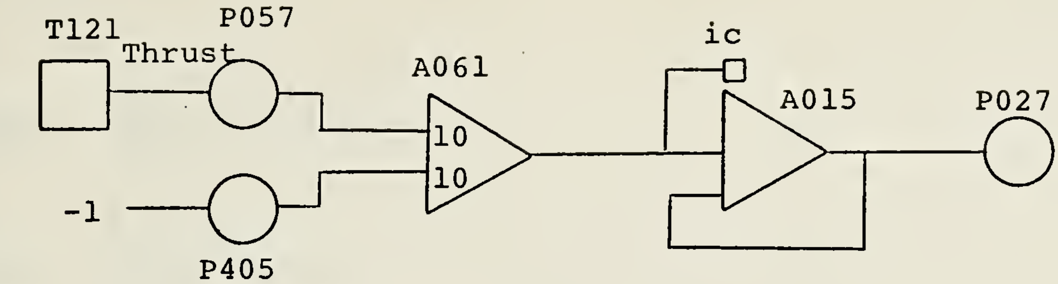

P016
p
P017
r
A014
P020
δa
10
10
P021
δr
P022
ϕ
T507
A013
v
P015
v
10
P023
r

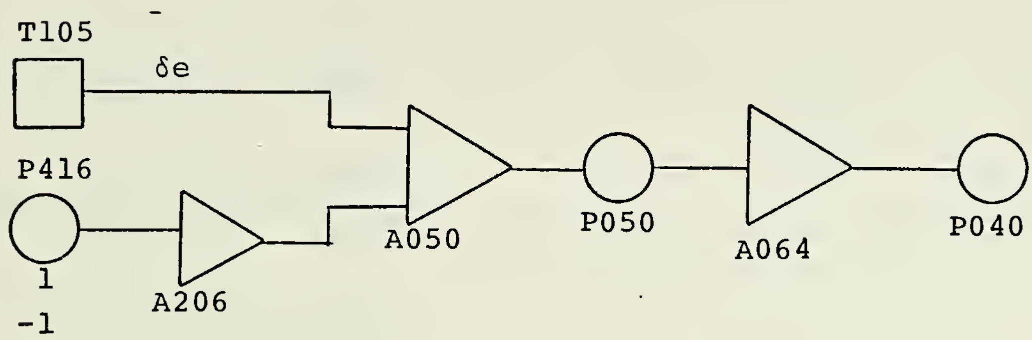

P025
w
P027
δ_T
A016
A017
u
P024
u
P026
θ

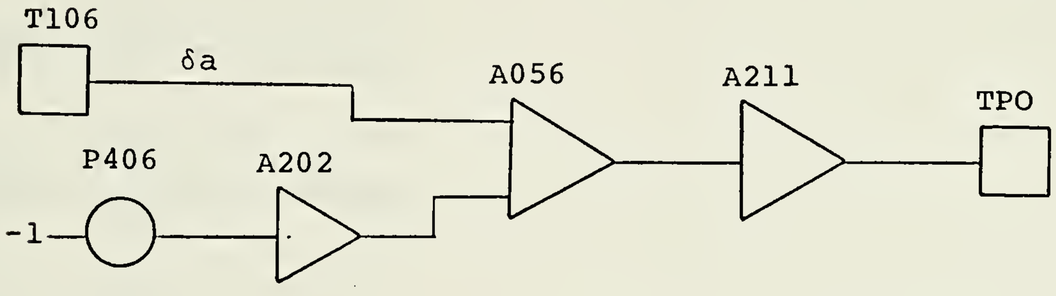

P030
u
P031
w
A021
A022
10
P033
q
P032
θ
$-\dot{w}$
A027
T503

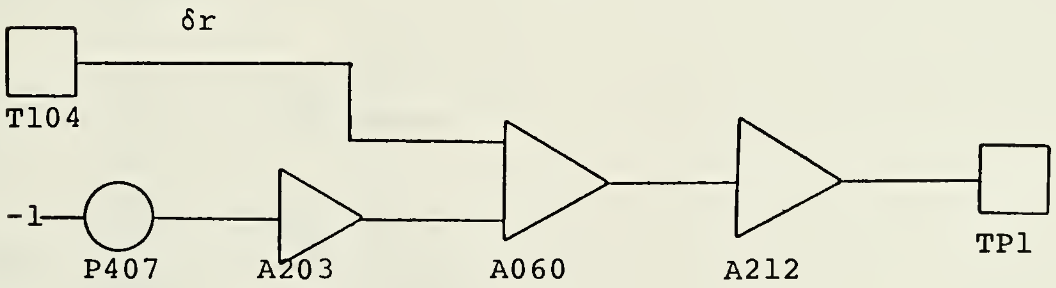

P034
u
P036
$-\dot{w}$
A024
A023
q
P035
w
10
P037
q
P040
δe


P054
A052
A003
T504
q
$-\dot{\theta}$
θ
P055
p
A007
T506
ϕ
P041
r
A076
A011
P042
r
$-\dot{\psi}$
ψ
T505
P413
+1
A026
A035
P044
u
T500
X'
P045
θ
P046
w
P047
A030
A031
+1
T501
$-\dot{y}'$
Y'
P056
v
P051
u
A025
A034
T502
P053
w
Z'
P417
P052
θ


T121
P057
Thrust
A061
10
10
ic
A015
P027
-1
P405


T105
δe
P416
A050
P050
A064
P040
1
-1
A206


T106
δa
A056
A211
TPO
P406
A202
-1


δr
T104
-1
P407
A203
A060
A212
TP1

$$P001 = \frac{P_{d_o} \; S \; b \; C_{\ell_\beta}}{I_{xx} \; u_o} \frac{SCV}{SCP}$$

$$P002 = \frac{P_{d_o} \; C_{\ell_p} \; S \; b^2}{2u_o \; I_{xx} \; (10)}$$

$$P003 = \frac{\dot{P}_{d_o} \; S \; b^2 \; C_{\ell_r}}{I_{xx} \; 2u_o} \frac{SCR}{SCP}$$

$$P004 = \frac{P_{d_o} \; S \; B \; C_{\ell_{\delta a}}}{I_{xx}} \frac{SCDA}{SCP}$$

$$P005 = \frac{P_{d_o} \; S \; b \; C_{\ell_{\delta r}}}{I_{xx}} \frac{SCDR}{SCP}$$

$$P006 = \frac{I_{xz}}{I_{xx}} \frac{SCR}{SCP}$$

$$P007 = \frac{P_{d_o} \; S \; b \; C_{n_\beta}}{I_{zz} \; u_o} \frac{SCV}{SCR}$$

$$P010 = \frac{P_{d_o} \; S \; b^2 \; C_{n_p}}{I_{zz} \; 2u_o} \frac{SCP}{SCR}$$

$$P011 = \frac{P_{d_o} \; S \; b^2 \; C_{n_r}}{I_{zz} \; 2u_o}$$

$$P012 = \frac{P_{d_o} \; S \; b \; C_{n_{\delta a}}}{I_{zz}} \frac{SCDA}{SCR}$$

$$P013 = \frac{P_{d_o} \; S \; b \; C_{n_{\delta r}}}{I_{zz}} \frac{SCDR}{SCR(10)}$$

$$P014 = \frac{I_{xz} \quad SCP}{I_{zz} \quad SCR}$$

$$P015 = \frac{P_{d_o} \; S \; C_{y_\beta}}{m \quad u_o}$$

$$P016 = \frac{P_{d_o} \; \bar{S} \; b \; C_{y_p} \quad SCP}{2 \quad m \; u_o \quad SCV}$$

$$P017 = \frac{P_{d_o} \; S \; b \; C_{y_r} \quad SCR}{2 \quad m \; u_o \quad SCV}$$

$$P020 = \frac{P_{d_o} \; S \; C_{y_{\delta a}} \quad SCDA}{m \quad SCV}$$

$$P021 = \frac{P_{d_o} \; S \; C_{y_{\delta r}} \quad SCDR}{m \quad SCV(10)}$$

$$P022 = \frac{P_{d_o} \; S \; C_{L_o} \quad SCPH}{m \quad SCV(10)}$$

$$P023 = \frac{- \; u_o \; SCR}{SCV \; (10)}$$

$$P024 = \frac{P_{d_o} \; S \; (-2) \; C_{D_o}}{m \quad u_o}$$

$$P025 = \frac{P_{d_o} \; S \; (C_{L_o} - C_{D_\alpha}) \; SCW}{m \quad u_o \quad SCU}$$

$$P026 = \frac{-P_{d_o} \; S \; C_{L_o} \quad SCTH}{m \quad SCU}$$

$$P027 = \frac{SCDT}{m \; SCU}$$

$$P030 = \frac{-2\ P_{d_o}\ S\ C_{L_o}}{m\ u_o}\ \frac{SCU}{SCW}$$

$$P031 = \frac{-\ (C_{L_\alpha} + C_{D_o})\ P_{d_o}\ S}{m\ u_o}$$

$$P032 = \frac{-\ C_{\bar{L}_o}\ P_{d_o}\ S\ (\tan\ \theta_o)}{m}\ \frac{SCTH}{SCW}$$

$$P033 = \frac{u_0\ SCQ}{SCW(10)}$$

$$P034 = \frac{P_{d_0}\ S\ \bar{c}\ C_{m_u}}{I_{yy}\ u_o}\ \frac{SCU}{SCQ}$$

$$P035 = \frac{P_{d_o}\ S\ \bar{c}\ C_{m_\alpha}}{I_{yy}\ u_o}\ \frac{SCW}{SCQ}$$

$$P036 = \frac{P_{d_o}\ S\ \bar{c}^2\ C_{m_{\dot{\alpha}}}}{2\ I_{yy}\ u_o^{\ 2}}\ \frac{SCW}{SCQ}$$

$$P037 = \frac{P_{d_o}\ S\ \bar{c}^2\ C_{m_q}}{2\ I_{yy}\ u_o}$$

$$P040 = \frac{P_{d_o}\ S\ \bar{c}\ C_{m_{\delta e}}}{I_{yy}}\ \frac{SCDE}{SCQ\ (10)}$$

$$P041 = \frac{\tan\ \theta_o\ SCR}{SCPH}$$

$$P042 = \frac{\sec\ \theta_o\ SCR}{SCSI}$$

$$P413 = \frac{u_o\ \cos\ \theta_o}{SCX}$$

$$P044 = \frac{\cos\theta_o \; SCU}{SCX}$$

$$P045 = \frac{-u_o \sin\theta_o \; SCTH}{SCX}$$

$$P046 = \frac{\theta_o \; SCW}{SCX}$$

$$P047 = \frac{u_o \cos\theta_o \; SCSI}{SCY}$$

$$P417 = \frac{-u_o \sin\theta_o}{SCZ}$$

P050 = .50

$$P051 = \frac{-\sin\theta_o \; SCU}{SCZ}$$

$$P052 = \frac{-u_o \cos\theta_o \; SCTH}{SCZ}$$

$$P053 = \frac{\cos\theta_o \; SCW}{SCY}$$

P054 = SCQ/SCTH

$$P055 = \frac{SCP}{(10)SCPH}$$

P056 = SCU/SCY

P057 = .50

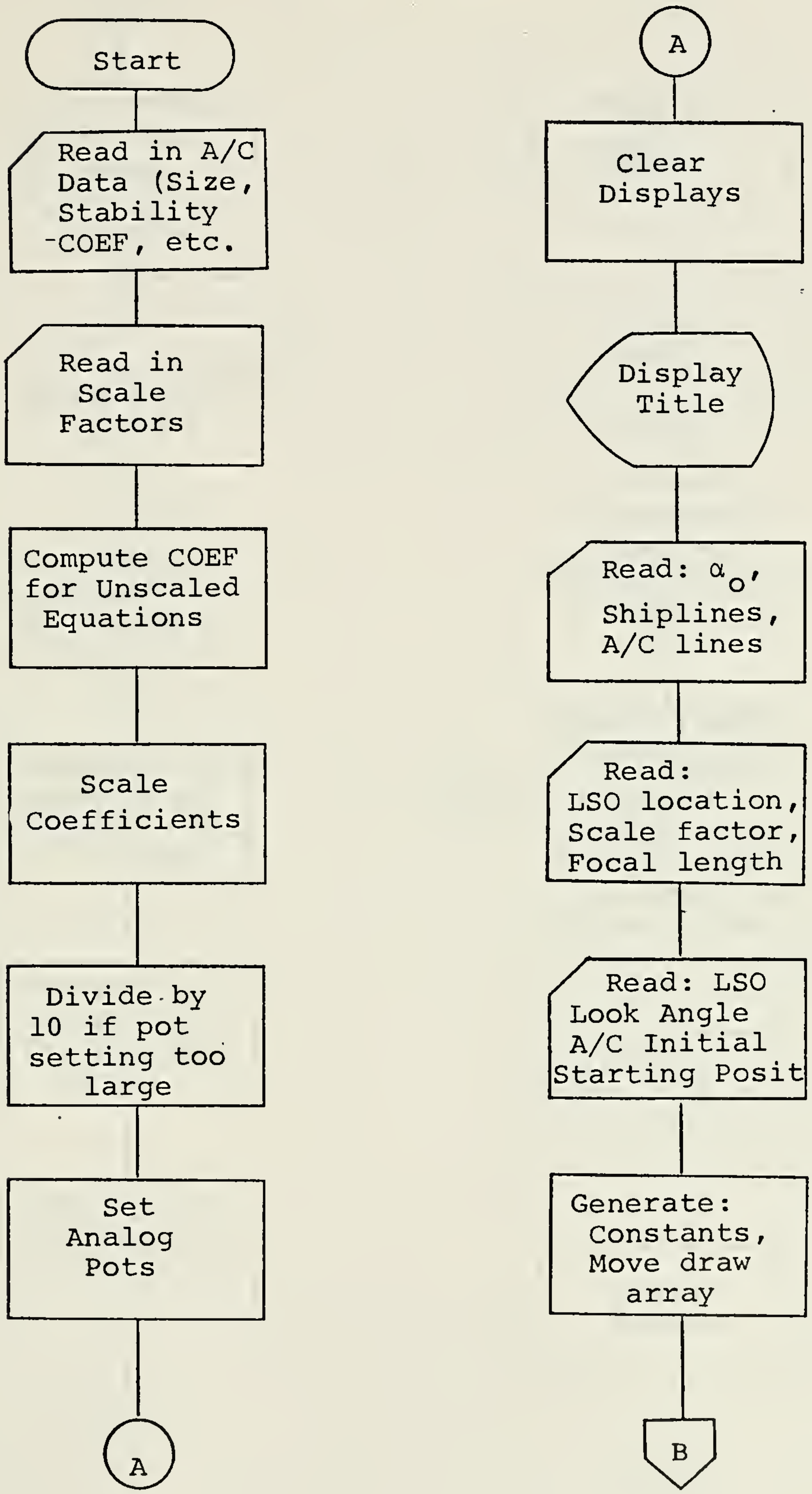
P405 = .1716 (BIAS)

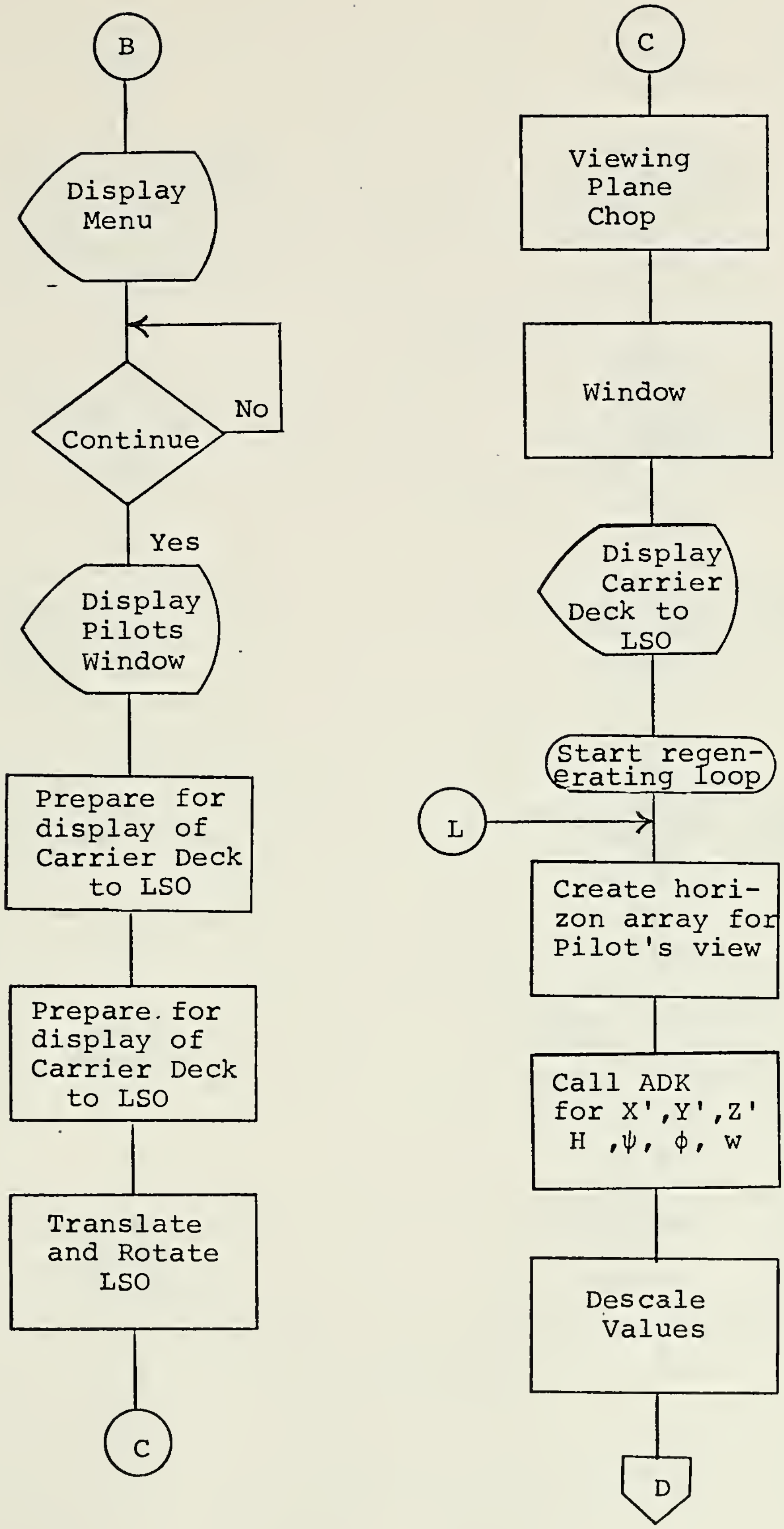
P406 = -.0576 (BIAS)

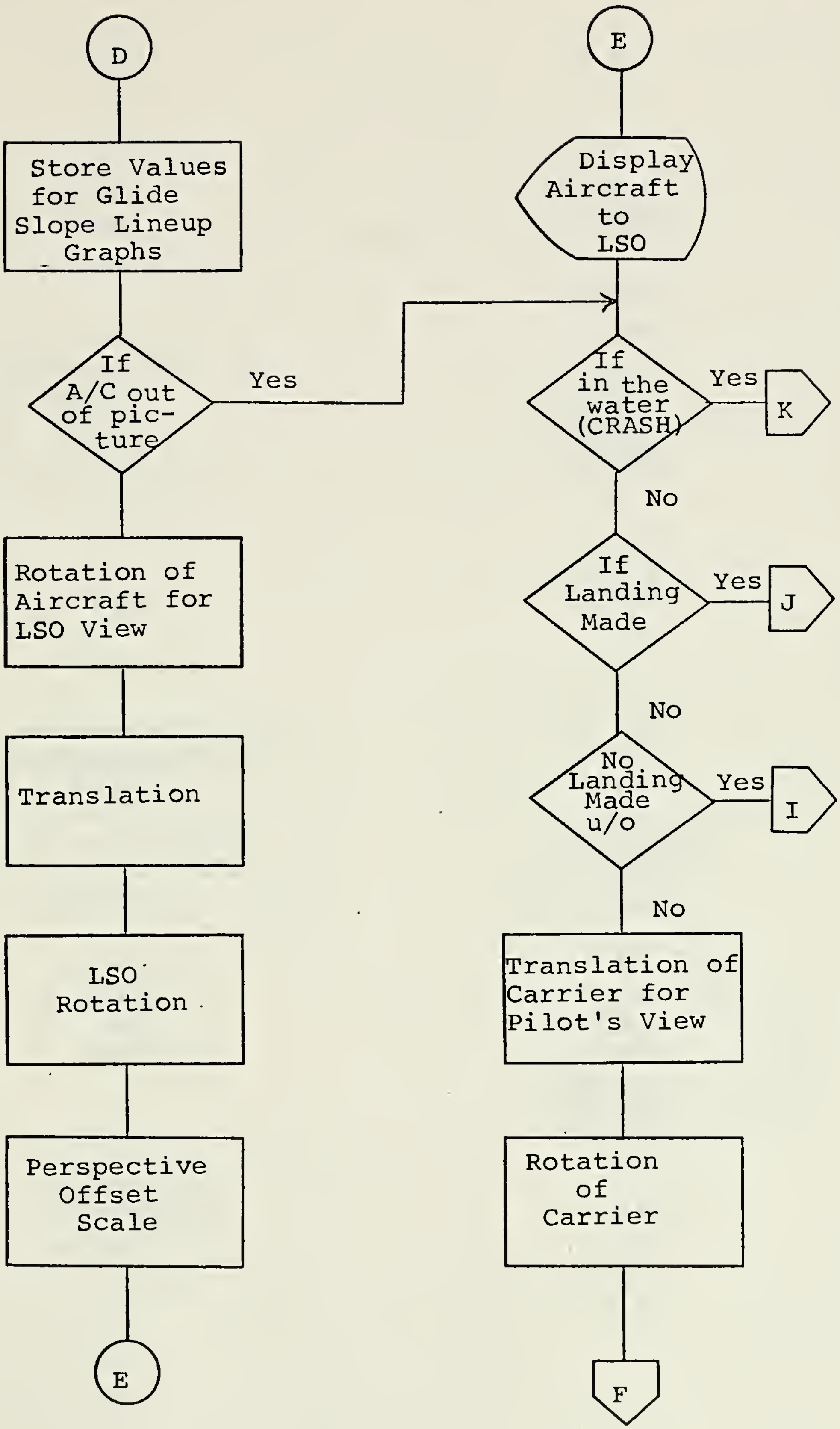
P407 = -.0963 (BIAS)

P416 = -.0594 (BIAS)

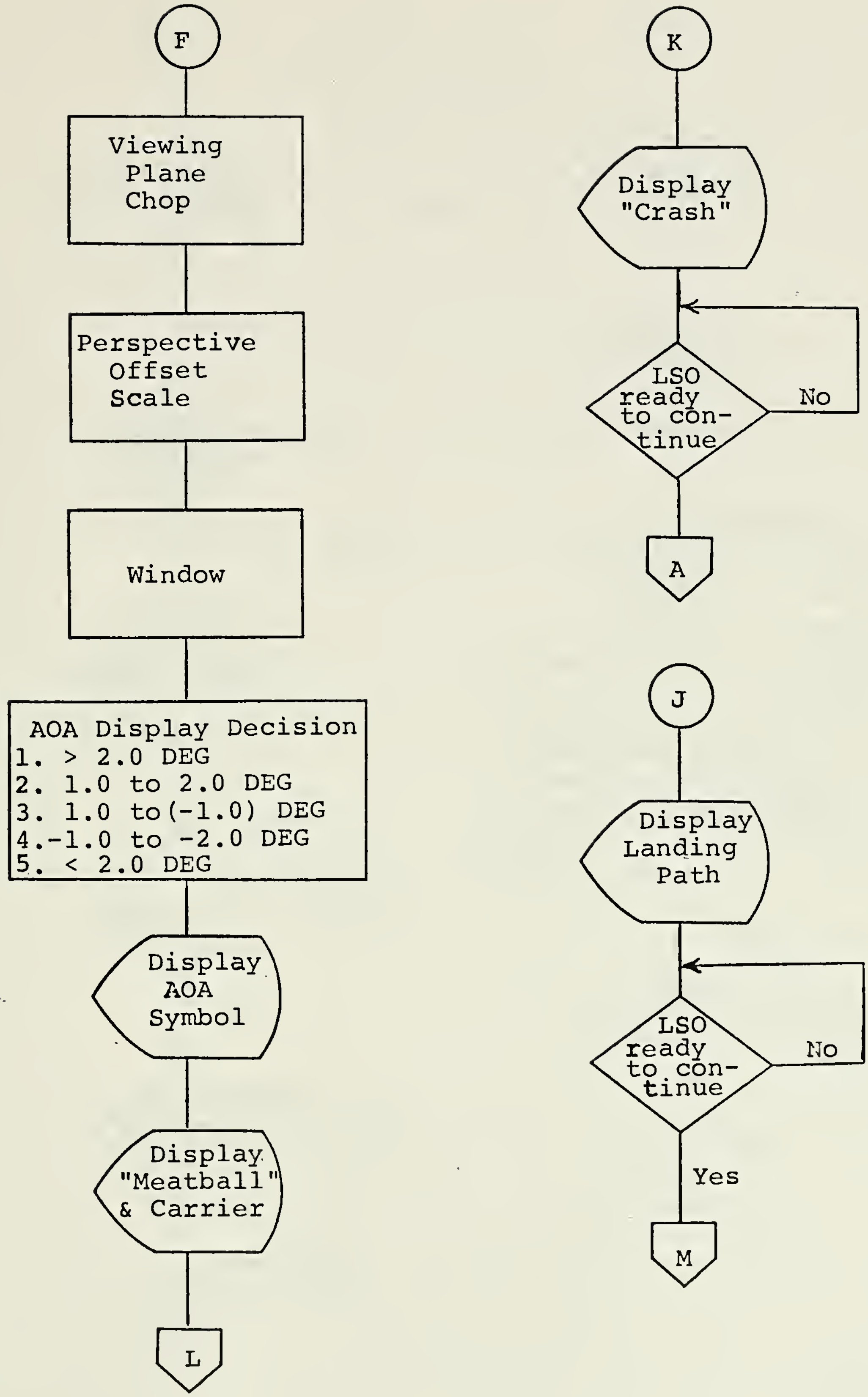
APPENDIX C. DIGITAL PROGRAM FLOW CHART

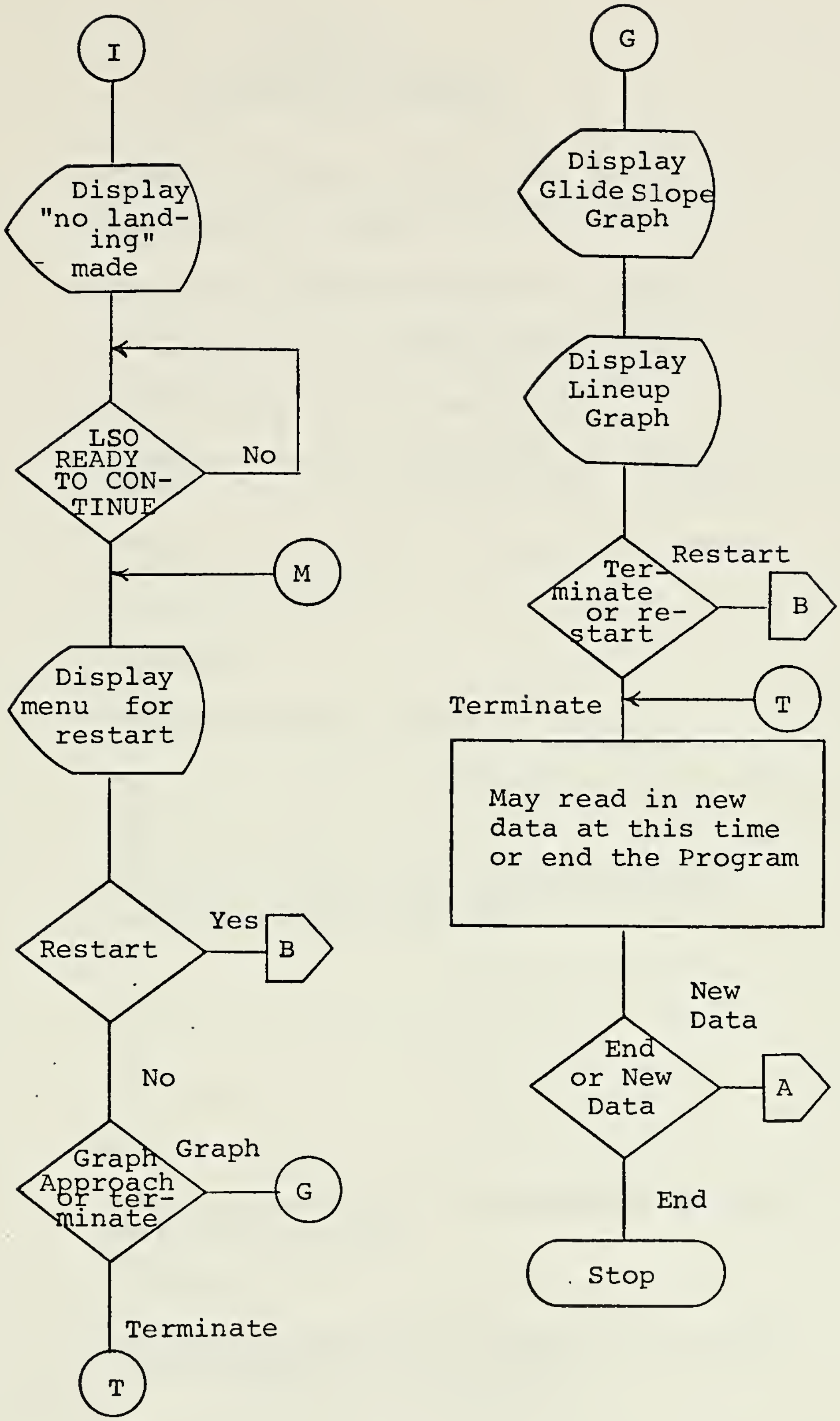



B
Display Menu
Continue
No
Yes
Display Pilots Window
Prepare for display of Carrier Deck to LSO
Prepare for display of Carrier Deck to LSO
Translate and Rotate LSO
C
C
Viewing Plane Chop
Window
Display Carrier Deck to LSO
Start regenerating loop
L
Create horizon array for Pilot's view
Call ADK for X',Y',Z' H ,ψ, φ, w
Descale Values
D



D
Store Values for Glide Slope Lineup Graphs
If A/C out of picture
Yes
Rotation of Aircraft for LSO View
Translation
LSO Rotation
Perspective Offset Scale
E
E
Display Aircraft to LSO
If in the water (CRASH)
Yes
K
No
If Landing Made
Yes
J
No
No Landing Made u/o
Yes
I
No
Translation of Carrier for Pilot's View
Rotation of Carrier
F


F
Viewing Plane Chop
Perspective Offset Scale
Window
AOA Display Decision
1. > 2.0 DEG
2. 1.0 to 2.0 DEG
3. 1.0 to(-1.0) DEG
4.-1.0 to -2.0 DEG
5. < 2.0 DEG
Display AOA Symbol
Display "Meatball" & Carrier
L
K
Display "Crash"
LSO ready to continue
No
A
J
Display Landing Path
LSO ready to continue
No
Yes
M


I
Display "no landing" made
LSO READY TO CONTINUE
No
M
Display menu for restart
Restart
Yes
B
No
Graph Approach or terminate
Graph
G
Terminate
T
G
Display Glide Slope Graph
Display Lineup Graph
Terminate or restart
Restart
B
Terminate
T
May read in new data at this time or end the Program
New Data
End or New Data
A
End
Stop

APPENDIX D

PROGRAM VARIABLES

AIXX	-	Moment of inertia about the X_B axis
AIXZ	-	Product of inertia
AIYY	-	Moment of inertia about the Y_B axis
AIZZ	-	Moment of inertia about the Z axis
ALPHA ϕ	-	Initial angle of attack (α_o)
AMBX	-	Array of vertical position of meatball lines
ANGL	-	Actual glide slope angle of the aircraft
AOA	-	Angle of attack
B	-	Wing span
CB	-	Mean aerodynamic chord ($\bar{c}$)
CDA	-	C_{D_α}
CD0	-	C_{D_0}
CHECK	-	X' distance from LSO to the starting position of the aircraft
CLA	-	C_{L_α}
CLO	-	C_{L_o}
CNST 1	-	$\alpha_0 + \theta_0$
CNST 2	-	CNST 1 + θ
CNST 3	-	Maximum position of the meatball before it goes off the lens
CNST 4	-	0.5/75.0
CNST 5	-	Distance from starting point of the A/C to the stern of the carrier
CTH0	-	cos (θ_o)

CYAW	- cos (Y)
DIFF	- Difference between actual glide slope angle and ideal glide slope angle
F	- Focal length for use in prospective matrix
FLG	- Flag which is set when aircraft is past the LSO
HINSCT	- Horizontal intersection of a line, used in the software window
I	- Integer variable, usually a counter
I1	- Number read from LSO's input to determine whether the restart, present a graph, or terminate option is wanted
I5	- First point in XKEEP array with a value less than 2000 ft
JAC	- Number of data cards needed to draw the aircraft
JAGN	- A counter used to check the number of times a point has gone through the software window
IBOGUS	- A packed, bogus array used to maintain constant loop time after the aircraft passed the LSO
IFAST	- Packed array for a fast AOA display
IGDIR<u>n</u>	- Graphics directory for AGT #<u>n</u>. Can use eight graphics blocks maximum
ISLO	- Packed array for a slow AOA display
IMBMD	- Move draw array for the meatball
ION	- Packed array for an on speed AOA display
IPIC	- Packed array with the fixed display shown to the LSO
IPLOT1	- Array of fixed coordinates for graphical glide slope or lineup display

IPLOT2	- Packed array to store actual glide slope track for graphing
IPLOT3	- Packed array to store actual lineup track
ISQ	- Packed array for fixed graphics shown to the pilot
ITDIRn	- Text directory for AGT #n. Maximum of 19 text blocks allowed
ITEST	- Decision variable to read in new data information or terminate the problem
ITEXnn	- Line of text sent from the digital computer to the graphics computer
IX/IY	- Operating values of start/end points used in software window
J	- Number of points to be plotted on the graphs
L	- Integer variable used as a counter
LL	- Counts once for every 10 regenerating loops
M	- Aircraft mass
MDC	- Move draw array for the runway and meat-ball presentation to the pilot
NAC	- Number of data cards necessary to describe the aircraft
NB	- 2* NL4 + 1
NL1	- NLSHIP + 1
NL2	- NLSHIP + 2
NL3	- NLSHIP + 3
NL4	- NLSHIP + 4
NLAND	- Number of lines needed to draw landing area as seen by the pilot. One line per data card
NLC	- NLAND + 1

NLC1 - 2 * NLC

NLC2 - 2 * NLC + 1

NLC5 - NLC1 + 5

NLC6 - NLC1 + 6

NLSHIP - Number of data cards needed to represent the lines which outline the carrier deck as seen by the LSO

ONEDEG - Limit for AOA display. Equals one degree in radians

ONEHAF - Limit for meatball display

P - Working matrix for aircraft stability derivatives

PHI - Scaled ϕ (change in roll angle from initial conditions)

PSI - Scaled ψ (change in yaw angle from initial conditions)

Q - Dynamic pressure

S - Wind area of the aircraft. Also used as the scale factor in the prospective matrix

SCDA - Scale factor for maximum aileron deflection

SCDE - Scale factor for maximum elevator deflection

SCDR - Scale factor for maximum rudder deflection

SCDT - Scale factor for maximum thrust

SCTH/SCPH/SCSI- Scale factors for maximum values of $\theta/\phi/$and ψ

SCU/SCV/SCW - Scale factors for maximum velocity change in the u/v/w directions

SCX/SCY/SCZ - Scale factors for maximum travel in the X'/Y'/Z' directions

SLOPE - Slope of display line, used in software window

STERN - Distance from the FRESNEL LENS to the Round-down

SYAW - sin (Y)

STHO - sin (THϕ)

TEMP1+2 - Temporary storage for a point on the horizon

THETA - θ (change in angle of climb from the initial condition)

THϕ - θ_ϕ (initial angle of climb)

TWODEG - Limit for AOA display. Equals two degrees in radians

uϕ - u_o (initial aircraft velocity)

VINSCT - Vertical intersection of a line used in the software window

W - Scaled change of velocity in the Z' direction

WIRE<u>n</u> - Distance from the FRESNEL LENS to the n^{th} wire x, n = 1,2,3,4

X/Y/Z - Scaled inertial X'/Y'/Z' positions of the aircraft from its starting position

X0/Y0/Z0 - Position on the carrier deck where the LSO's eye is located (measured in the X_c, Y_c, Z_c axis system [Fig. 5])

XAC/YAC/ZAC - Coordinate system in which the aircraft is drawn (X_B,Y_B,Z_B in Fig. 5)

XAC<u>n</u>/YAC<u>n</u>/ZAC<u>n</u> - Working vectors for aircraft data

XE<u>n</u>/YE<u>n</u>/ZE<u>n</u> - Working vectors to handle carrier deck points

XELND/YELND/ZELND - End point for lines that represent the landing area as seen by the pilot

XESHIP/YESHIP/ZESHIP - Working vectors used to process carrier deck graphics as seen by the LSO

XESHP/YESHP/ZESHP - End point of lines describing the carrier deck

XINIT/ZINIT - Starting position for the aircraft (measured in X_c, Y_c, Z_c axis system with $Y_c = 0.0$)

XKEEP/YKEEP/ZKEEP - Array holding inertial X/Y/Z position of the aircraft for use in the glide slope and lineup graphs

XSn/YSn/ZSn - Working vectors used to handle carrier deck points

XSLND/YSLND/ZSLND - Starting point for lines used to draw the landing area as seen by the pilot

XSSHIP/YSSHIP/ZSSHIP - Working arrays used to process carrier deck graphics as seen by the LSO

XSSHP/YSSHP/ZSSHP - Starting point of a line used to describe the carrier deck as seen by the LSO

XT/YT/ZT - Inertial distance from the LSO to the aircraft (X',Y',Z')

XT1/YT1/ZT1 - Inertial distance from the pilot to the origin of the X_c, Y_c, Z_c axis [Fig. 5]

Y - Value used in software window, also used as the scaled Y' position of the aircraft

YMB - Array containing the horizontal position of the displayed meatball

YTEMP - Temporary storage for Y-coordinate in software window

YUPLSO - Yaw angle of the viewing plane of the LSO

Z - Value used in software window, also the scaled inertial Z' position of the aircraft

ZINIT1 - Distance from aircraft starting point to the water

ZSAVE - Save the value of Z because of its dual use in the window

APPENDIX E

OPERATING INSTRUCTIONS

The program exists in two forms. A card deck which must be input through the card reader is available in the computer room in a box marked "Pilot/LSO/aircraft interaction." The program is also located on a tape marked in the same manner as the box. This tape is a core dump and doesn't give a listing when used. Use of the tape requires about 15-20 minutes less to set up.

Operating instructions for the various computers is located in the computer room on the fifth floor of Spanagel Hall.

1. Load the tape marked "Pilot/LSO/aircraft interaction" or place the large deck of cards in the card reader.

2. On the Tape Control Panel: Put tape at the load point, select automatic, set unit select to #1.

3. If the large card deck is used, mount a scratch tape and omit step 4.

4. Put the card deck marked "For use with tape" in the card reader. Ready the card reader.

5. Push IDLE, RESET, RUN, CARDS on the XDS 9300 control panel.

6. Mount analog and logic boards numbered 21 onto the Cl 5000.

7. Press "Digital Computer" on the analog control panel. The only lights that should be on are LOCAL, POTSET, IDC-X1, REAL TIME, and CLOCK1-H.

8. Seven pots must be set manually. A list of these pots and their values should be listed on the XDS 9300 teletype. Set the seven pots. Return the analog to "Digital Control."

9. Set the pilot's chair in front of the AGT #2.

10. Ready both AGT units and set in GATD1 by typing:

RESET ("GATD1", 104)!

GATED!

If "FILE NOT FOUND" appears, type 04 vice 104.

11. The program asks the operator to type a one (1) on the XDS 9300 teletype "to continue." The program will have to be reloaded if both AGT units are not ready when a 1 is typed. Type a 1 and a carriage return on the XDS 9300 teletype.

12. Text should appear on both AGT scopes. At this point the AGT-1 teletype has control of the problem. Follow the instructions on the scope.

13. After an approach has been completed, the LSO has three options:

a. Begin a new approach.

b. Have displayed a graphical output of glide scope and lineup results from the approach.

c. Terminate (i.e. give control back to the proctor).

If terminate is selected, the following will appear on the

XD5-9300 teletype: "Type 1 to Read new data, Type 0 to Stop". The proctor, at this point, may change any data card and place the data deck (without the control cards) in the card reader. Then type a 1 on the XDS-9300 teletype and go to step 11.

14. The program is ended by typing a zero instead of a one on the XDS 9300 teletype.

APPENDIX F

PREPARATION OF DATA DECK

Card 1-7

The first seven cards are used to read in non-dimensional aircraft stability derivatives and other aircraft parameters. Ten spaces are provided for each number. Slots 1-10 are used for the first number on a card, 11-20 for the second number, etc. The first seven cards and their contents are listed below. Note: All non-dimensional derivatives are per radian except as noted.

1. 5(ft), 6(ft), $\bar{c}$(ft), I_{xx}(slug-ft^2), I_{yy}(slug-ft^2), I_{zz}(slug-ft^2), I_{xz}(slug-ft^2)
2. C_{ℓ_β}, C_{ℓ_p}, C_{ℓ_r}, $C_{\ell_{\delta a}}$ (per deg), $C_{\ell_{\delta r}}$ (per deg)
3. C_{n_β}, C_{n_p}, C_{n_r}, $C_{n_{\delta a}}$ (per deg), $C_{n_{\delta r}}$ (per deg)
4. C_{y_β}, C_{y_p}, C_{y_r}, $C_{y_{\delta a}}$ (per deg), $C_{y_{\delta r}}$ (per deg)
5. C_{m_u}, C_{m_α}, $C_{m_{\dot\alpha}}$, C_{m_q}, $C_{m_{\delta e}}$ (per deg)
6. C_{L_α}, C_{D_o}, C_{D_α}, C_{L_o}, u_o (ft/sec)
7. θ_o, A/C mass (slugs)

Note: Use a decimal point with all numbers.

Card 8&9

The next two cards contain scaling values. Again, ten spaces are provided for each number. Use scale values for

the following variables:

8. δr, δe, δa, δ_T, u, v, w, θ

9. ϕ, ψ, X', Y', Z', p, q, r

10. ALPHA zero in the first 10 spaces.

11. The next set of cards describe the lines which represent the carrier deck seen by the LSO. The first card contains the number of lines needed to draw the ship (NLSHIP-I2). NLSHIP is the number of cards that follow since each line used to draw the ship uses one card.

The next NLSHIP cards are prepared using the X_c, Y_c, Z_c axis system [Fig. 5] with its origin fixed to the carrier deck as the point where the center line of the angled deck meets a line perpendicular to the center line and going through the FRESNEL LENS. The positive X_c axis points toward the bow of the ship, the positive Y_c axis points toward the port side of the ship.

Expressed in the X_c, Y_c, Z_c axis system, the lines describing the carrier deck are put onto cards as follows:

Columns 1-10, 11-20, 21-30 the coordinates of the beginning of a line.

Columns 31-40, 41-50, 51-60 the coordinates of the end of the same line.

The next set of cards outline the aircraft. The first card contains the number of cards needed to describe the aircraft (NAC). Format I2

The axis system used is (X_B, Y_B, Z_B) [Fig. 5] with the origin at the center of gravity of the aircraft. The

positive X_B axis points toward the nose of the A/C, the positive Y_B axis is out the left wing and the positive Z_B axis is up. In this axis system, put the coordinates of all intersecting lines in column 1-10 (X_B), 11-20 (Y_B), and 21-30 (Z_B). In column 31 place a zero to move and a 1 to draw a vector.

Maximum number of lines is 60.

The next card contains the number of cards following which describe the landing area as seen by the pilot. Formal I2. The program had five lines drawn and represent a rectangular runway 1000 ft. long and 100 ft. wide. It is recommended that these cards remain the same since more lines would increase the loop time. If these cards are replaced, the format and axis system is the same as the one used in describing the carrier deck as seen by the LSO.

The next card contains in: columns 1-10, 11-20, 21-30, the position of the LSO's eye expressed in the X_C, Y_C, Z_C axis system; in column 31-40 the scale factor (S) equal to 1.0; in column 41-50 the focal length (F) equal to 20 feet.

The next card contains the number of degrees the LSO moves his head and this his field of view. A positive angle is equivalent to moving his head to the left. Use column 1-10.

The next card: Column 1-10, 11-20 the X_C and Y_C starting position of the aircraft. (XINIT and YINIT)

The last card contains in column 1-10, 11-20, 21-30, and 31-40 the distance between the FRESNEL LENS and the #1, #2, #3, #4 wire respectively. Columns 41-50 contains the distance from the FRESNEL LENS to the stern of the ship.

Wire 1 Wire 2 Wire 3 Wire 4 Stern

KINIT YINIT

YVPLSO

XO YO ZO S F

NLAND

NLAND→NUMBER OF CARDS TO FOLLOW WHICH DESCRIBE THE PILOT VIEWED LANDING AREA

NAC

NAC→NUMBER OF CARDS TO FOLLOW WHICH DESCRIBE THE AIRCRAFT

NLSHIP

11 NLSHIP→NUMBER OF CARDS TO FOLLOW WHICH DESCRIBE THE SHIP (I2)

10 α_o

8&9 SCALING PARAMETERS FOR ANALOG COMPUTERS

7. θ_o, A/C MASS (SLUGS).

1-6 A/C CONSTANTS

Data Deck Sample

APPENDIX G. COMPUTER PROGRAM

```
***********************************************************************
*                                                                     *
*       PILOT / CARRIER / AIRCRAFT INTERACTION                        *
*                           BY                                        *
*                  LT. P.G. STUECK                                    *
*                           AND                                       *
*                  LTJG M.H. REDLIN                                   *
*                                                                     *
***********************************************************************
*                                                                     *
      DIMENSION IGDIR1(9),IAC(50),XAC(50),YAC(50),ZAC(50),XAC1(50),
     1YAC1(50),ZAC1(50),XAC2(50),YAC2(50),ZAC2(50),XAC3(50),
     2YAC3(50),ZAC3(50),XAC4(50),YAC4(50),ZAC4(50),XSSHIP(50),
     3YSSHIP(50),ZSSHIP(50),XESHIP(50),YESHIP(50),ZESHIP(50),
     4XS1(50),YS1(50),ZS1(50),XE1(50),YE1(50),ZE1(50),XS2(50),
     5YS2(50),ZS2(50),XE2(50),YE2(50),ZE2(50),IPIC(50),MD(50),
     6XSLND(50),YSLND(50),ZSLND(50),XELND(50),YELND(50),ZELND(50),
     7MDC(50),ILAND(50),ISQ(100),IFAST(10),ION(10),ISLO(10),
     8AMBX(10),AMBY(10),IMBMD(10),IMBLU(10),ITDIR1(20),ITEX1(7),
     9ITEX2(7),ITEX3(1),ITEX4(4),ITEX5(1),ITEX6(3),ITEX7(1),
     1IGDIR2(9),ITDIR2(20),
     2ITEX9(9),ITEX10(8),ITEX11(8),ITEX12(3),ITEX13(8),ITEX14(9),
     3ITEX15(8),ITEX16(9),ITEX17(8),ITEX18(3),ITEX19(9),ITEX20(8),
     4ITEX21(8),ITEX22(9),ITEX23(2),ITEX24(2),ITEX25(6),ITEX26(4),
     5ITEX27(9),ITEX28(9),ITEX29(5),ITEX8(5),XSSHP(50),YSSHP(50),
     6ZSSHP(50),XESHP(50),YESHP(50),ZESHP(50),IBCGUS(50),
     7XKEEP(100),YKEEP(100),ZKEEP(100),IPLOT1(100),IPLOT2(100),
     8IPLOT3(100),ITEX30(9),ITEX31(9),ITEX32(8),ITEX33(1),ITEX34(1),
     9ITEX35(1),ITEX36(1),ITEX37(1),ITEX38(1),ITEX39(1),ITEX40(1),
     1ITEX41(1),ITEX42(1),ITEX43(1),ITEX44(5),ITEX45(4),ITEX46(9),
     2ITEX47(4),ITEX48(4),ITEX49(4),ITEX50(4),ITEX51(10),ITEX52(3),
     3PCT(46),P(4,5),ITEX53(7),ITEX54(10),ITEX55(10),ITEX56(2)
      DATA NULL/77777777B/
  100 FORMAT(7F10.2)
  101 FORMAT(5F10.2)
  102 FORMAT(2F10.2)
  103 FORMAT('1','COEFFICIENTS'//)
  104 FORMAT(' ','POT(',I2,')=',F11.6)
  105 FORMAT(8F10.2)
  106 FORMAT('0','CL-BETA=',E13.5,/,' CL-P=',E13.5,/,' CL R=',E13.5,/,
     1' CL-DELTA-A=',E13.5,/,' CL-DELTA-R=',E13.5,/,' CN-BETA',E13.5,/,
     2' CN-P=',E13.5,/,' CN-R=',E13.5,/,' CN-DELTA-A=',E13.5,/,
     3' CN-DELTA-R=',E13.5,/,' CY-BETA=',E13.5,/,' CY-P=',E13.5,/,
     4' CY-R=',E13.5,/,' CY-DELTA-A=',E13.5,/,' CY-DELTA-R=',E13.5,/,
     5' CM-U=',E13.5,/,' CM-ALPHA=',E13.5,/,' CM-ALPHA-DOT=',E13.5,/,
     6' CM-Q=',E13.5,/,' CM-DELTA-E=',E13.5,//)

  107     FORMAT('0',//,'  SET POT P413 TO THE VALUE OF PCT(43)',//,
```

```
    1'  SET POT P417 TO THE VALUE OF POT(50)')
108 FORMAT('0','DIVIDED POT(',I2,') BY 10.0')
109 FORMAT('1','SCALED COEFFICIENTS'//)
110 FORMAT('1','POT SETTINGS '//)
111 FORMAT('1','INPUT CHECK'//)
112 FORMAT('SET POT 413 TO ',F5.4)
113 FORMAT('SET POT 417 TO ',F5.4)
200 FORMAT(I1)
201 FORMAT(I2)
202 FORMAT(6F10.2)
203 FORMAT(5F10.2)
204 FORMAT(3F10.2)
205 FORMAT(F10.2)
206 FORMAT(E20.6)
207 FORMAT(I4)
208 FORMAT(3F10.2,I1)
209 FORMAT('PILOT - LSO INTERACTION      ')
210 FORMAT('CARRIER LANDING SIMULATION   ')
211 FORMAT('BY  ')
212 FORMAT('LT P.G. STUECK  ')
213 FORMAT('AND ')
214 FORMAT('M.H. REDLIN ')
215 FORMAT('    ')
216 FORMAT('PUSH C/R TO CONTINUE')
217 FORMAT('THIS PROGRAM IS UNDER THE CONTROL    ')
218 FORMAT('OF THE LSO. HE WILL LET YOU KNOW')
219 FORMAT('WHEN HE IS READY TO HAVE YOU FLY')
220 FORMAT('THE AIRCRAFT')
221 FORMAT('THIS PROGRAM IS DESIGNED TO TEST')
222 FORMAT('THE PILOT/LSO/AIRCRAFT INTERACTION. ')
223 FORMAT('PARAMETER CHANGES WILL BE THE   ')
224 FORMAT('RESPONSIBILITY OF THE PROCTOR AND    ')
225 FORMAT('THESE CHANGES HAVE BEEN MADE BY ')
226 FORMAT('THIS TIME.  ')
227 FORMAT('TO CONTINUE INFORM THE PILOT THAT    ')
228 FORMAT('YOU ARE READY FOR AN APPROACH.  ')
229 FORMAT('A CARRAGE RETURN WILL START THE ')
230 FORMAT('APPROACH AT ABOUT 1 MILE FROM THE    ')
231 FORMAT('CARRIER.')
232 FORMAT('CRASH   ')
233 FORMAT('LANDING DATA                ')
234 FORMAT('NO LANDING MADE ')
235 FORMAT('TYPE A C/R TO START A NEW APPROACH. ')
236 FORMAT('TYPE 2. (DONT FORGET THE DECIMAL)    ')
237 FORMAT('AND C/R TO TERMINATE      ')
238 FORMAT('TYPE   1. (DONT FORGET THE DECIMAL) ')
239 FORMAT('AND A C/R TO GET A GRAPHICAL OUTPUT ')
240 FORMAT('OF THE AIRPLANES ACTUAL TRACK     ')
```

```
241 FORMAT(' 50 ')
242 FORMAT('-50 ')
243 FORMAT('-100')
244 FORMAT('100 ')
245 FORMAT('200 ')
246 FORMAT('300 ')
247 FORMAT('400 ')
248 FORMAT('500 ')
249 FORMAT('1000')
250 FORMAT('1500')
251 FORMAT('2000')
252 FORMAT('GLIDE SLOPE PROFILE ')
253 FORMAT('LINE UP PROFILE ')
254 FORMAT('(ALL DISTANCES MEASURED IN FEET)      ')
255 FORMAT('(LINEUP LEFT)    ')
256 FORMAT('(LINEUP RIGHT)   ')
257 FORMAT('BOLTER-BOLTER    ')
258 FORMAT('WIRE NUMBER ',I1,'    ')
259 FORMAT('LINE UP ',F5.2,'FEET RIGHT OF CENTERLINE    ')
260 FORMAT('LINE UP ',F5.2,'FEET LEFT  OF CENTERLINE    ')
261 FORMAT('RAMP-STRIKE ')
262 FORMAT('CRASH DUE TO AN EXCESSIVE    ')
263 FORMAT('RATE OF DESCENT OF  ',F5.0,'FEET PER MINUTE ')
264 FORMAT('RATE OF DESCENT IS  ',F4.0,' FEET PER MINUTE ')
265 FORMAT('(RAMP)  ')
551 READ(5,100) S,B,CB,AIXX,AIYY,AIZZ,AIXZ
    READ(5,101)((P(I,J),J=1,5),I=1,4)
    READ(5,101) CLA,CDO,CDA,CLO,U0
    READ(5,102)THO,M
    READ(5,105) SCDR,SCDE,SCDA,SCDT,SCU,SCV,SCW,SCTH
    READ(5,105) SCPH,SCSI,SCX,SCY,SCZ,SCP,SCQ,SCR
    WRITE(6,111)
    OUTPUT(6) S,B,CB,AIXX,AIYY,AIZZ,AIXZ
    WRITE(6,106)((P(I,J),J=1,5),I=1,4)
    OUTPUT(6) CLA,CDO,CDA,CLO,U0,THO,M
    OUTPUT(6) SCDR,SCDE,SCDA,SCDT,SCU,SCV,SCW,SCTH,SCPH,SCSI,SCX,
   1SCY,SCZ,SCP,SCQ,SCR
    THO=THO/57.296
    B1=B/(2.0*U0)
    CB1=CB/(2.0*U0**2)
    Q=.5*.002378*U0**2*S
    CTHO=COS(THO)
     STHO=SIN(THO)
    Q1=(Q*B)/(AIXX)
    Q2=(Q*B)/AIZZ
    Q3=Q/M
    Q4=(Q*CB)/AIYY
COMPUTE  COEFFICIENTS
```

```
      POT(1)=Q1*P(1,1)/UO
      POT(2)=Q1*P(1,2)*B1
      POT(3)=Q1*P(1,3)*B1
      POT(4)=Q1*P(1,4)
      POT(5)=Q1*P(1,5)
      POT(6)=AIXZ/AIXX
      POT(7)=Q2*P(2,1)/UO
      POT(8)=Q2*P(2,2)*B1
      POT(9)=Q2*P(2,3)*B1
      POT(10)=Q2*P(2,4)
      POT(11)=Q2*P(2,5)
      POT(12)=AIXZ/AIZZ
      POT(13)=Q3*P(3,1)/UO
      POT(14)=Q3*P(3,2)*B1
      POT(15)=Q3*P(3,3)*B1
      POT(16)=Q3*P(3,4)
      POT(17)=Q3*P(3,5)
      POT(18)=CLO*Q3
      POT(19)=-UO
      POT(20)=-(Q3*2.0*CDO    )/UO
      POT(21)=Q3*(CLO-CDA)/UO
      POT(22)=-CLO*Q3
      POT(23)=1.0/M
      POT(24)=-(Q3*2.0*CLO)/UO
      POT(25)=-Q3*(CDO+CLA)/UO
      POT(26)=-(Q3*CLO*STHO)/CTHO
      POT(27)=UO
      POT(28)=Q4*P(4,1)/UO
      POT(29)=Q4*P(4,2)/UO
      POT(30)=Q4*P(4,3)*CB1
      POT(31)=Q4*P(4,4)*CB1  *UO
      POT(32)=Q4*P(4,5)
      POT(33)=STHO/CTHO
      POT(34)=1.0/CTHO
      POT(35)=UO*CTHO
      POT(36)=CTHO
      POT(37)=-UO*STHO
      POT(38)=THO
      POT(39)=UO*CTHO
      POT(40)=-UO*STHO
      POT(41)=-STHO
      POT(42)=-UO*CTHO
      POT(43)=CTHO
      WRITE(6,103)
      DO 70   I=1,43
      WRITE(6,104) I,POT(I)
   70 CONTINUE
SCALE  COEFFICIENTS
```

```
      POT(1)=POT(1)*SCV/SCP
      POT(3)=POT(3)*SCR/SCP
      POT(4)=POT(4)*SCDA/SCP
      POT(5)=POT(5)*SCDR/SCP
      POT(6)=POT(6)*SCR/SCP
      POT(7)=POT(7)*SCV/SCR
      POT(8)=POT(8)*SCP/SCR
      POT(10)=POT(10)*SCDA/SCR
      POT(11)=POT(11)*SCDR/SCR
      POT(12)=POT(12)*SCP/SCR
      POT(14)=POT(14)*SCP/SCV
      POT(15)=POT(15)*SCR/SCV
      POT(16)=POT(16)*SCDA/SCV
      POT(17)=POT(17)*SCDR/SCV
      POT(18)=POT(18)*SCPH/SCV
      POT(19)=POT(19)*SCR/SCV
      POT(21)=POT(21)*SCW/SCU
      POT(22)=POT(22)*SCTH/SCU
      POT(23)=POT(23)*SCDT/SCU
      POT(24)=POT(24)*SCU/SCW
      POT(26)=POT(26)*SCTH/SCW
      POT(27)=POT(27)*SCQ/SCW
      POT(28)=POT(28)*SCU/SCQ
      POT(29)=POT(29)*SCW/SCQ
      POT(30)=POT(30)*SCW/SCQ
      POT(32)=POT(32)*SCDE/SCQ
      POT(33)=POT(33)*SCR/SCPH
      POT(34)=POT(34)*SCR/SCSI
      POT(35)=POT(35)*1.0/SCX
      POT(36)=POT(36)*SCU/SCX
      POT(37)=POT(37)*SCTH/SCX
      POT(38)=POT(38)*SCW/SCX
      POT(39)=POT(39)*SCSI/SCY
      POT(40)=POT(40)*1.0/SCZ
      POT(41)=POT(41)*SCU/SCZ
      POT(42)=POT(42)*SCTH/SCZ
      POT(43)=POT(43)*SCW/SCZ
      POT(44)=SCQ/SCTH
      POT(45)=SCP/SCPH
      POT(46)=SCV/SCY
MUTTIPLY BY TEN ON ANALOG BOARD
      POT(11)=POT(11)*0.1
      POT(17)=POT(17)*0.1
      POT(32)=POT(32)*0.1
      WRITE(6,109)
      DO 71   I=1,46
      WRITE(6,104) I,POT(I)
   71 CONTINUE
```

```
      DO 72 I=1,46
      IF(ABS(POT(I)).GT.1)    WRITE(6,108) I
      IF(ABS(POT(I)).GT.1) POT(I)=0.1*POT(I)
      POT(I)=ABS(POT(I))
   72 CONTINUE
      WRITE(6,110)
      DO 73 I=1,46
      K=I+((I/8)*2)
      WRITE(6,104) K,POT(I)
   73 CONTINUE
      WRITE(6,107)
C SET ANALOG POTS
      CALL SETPOT(4HP001,POT(1),4HP002,POT(2),4HP003,POT(3),4HP004,
     1POT(4),4HP005,POT(5),4HP006,POT(6),4HP007,POT(7),4HP010,POT(8),
     24HP011,POT(9),4HP012,POT(10),4HP013,POT(11),4HP014,POT(12),
     34HP015,POT(13),4HP016,POT(14),4HP017,POT(15),4HP020,POT(16),
     44HP021,POT(17),4HP022,POT(18),4HP023,POT(19),4HP024,POT(20),
     54HP025,POT(21),4HP026,POT(22),4HP027,POT(23),4HP030,POT(24),
     64HP031,POT(25),4HP032,POT(26),4HP033,POT(27),4HP034,POT(28),
     74HP035,POT(29),4HP036,POT(30),4HP037,POT(31),4HP040,POT(32),
     84HP041,POT(33),4HP042,POT(34),4HP044,POT(36),
     94HP045,POT(37),4HP046,POT(38),4HP047,POT(39),4HP051,POT(41),
     14HP052,POT(42),4HP053,POT(43),4HP054,POT(44),4HP055,POT(45),
     24HP056,POT(46),4HP000,.1000,4HP057,.5000)
C ENCODE MENUE
      ENCODE(28,209,ITEX1)
      ENCODE(28,210,ITEX2)
      ENCODE(04,211,ITEX3)
      ENCODE(16,212,ITEX4)
      ENCODE(04,213,ITEX5)
      ENCODE(12,214,ITEX6)
      ENCODE( 4,215,ITEX7)
      ENCODE(20,216,ITEX8)
      OUTPUT(101) ' '
      OUTPUT(101) 'SET THRUST BIAS POT 405 TO 1716 '
      OUTPUT(101) 'SET AILERON BIAS POT 406 TO -0576 '
      OUTPUT(101) 'SET RUDDER BIAS POT 407 TO -0963 '
      WRITE(101,112) POT(35)
      OUTPUT(101) 'SET ELEVATOR BIAS POT 416 TO -0594 '
      WRITE(101,113) POT(40)
      OUTPUT(101) 'SET ELEVATOR GAIN POT 050 TO 1000 '
      OUTPUT(101)'THIS IS AN AGT TRAP. TYPE 1 TO CONTINUE'
      READ(101,200) TRAP
      CALL DGINIT(    1,IGDIR1, 9,IER)
      CALL DGINIT(    2,IGDIR2, 9,IER)
      CALL DTINIT(1,ITDIR1,20,IER)
      CALL DTINIT(2,ITDIR2,20,IER)
      CALL TEXTO(    1 ,ITEX1,7, 4,10,3,3,IER)
```

```
      CALL TEXTO(   2 ,ITEX1,7, 4,10,3,3,IER)
      CALL TEXTO(   1 ,ITEX2,7,10,10,3,3,IER)
      CALL TEXTO(   2 ,ITEX2,7,10,10,3,3,IER)
      CALL TEXTO(   1 ,ITEX3,1,20,33,2,3,IER)
      CALL TEXTO(   2 ,ITEX3,1,20,33,2,3,IER)
      CALL TEXTO(   1 ,ITEX4,4,24,26,2,3,IER)
      CALL TEXTO(   2 ,ITEX4,4,24,26,2,3,IER)
      CALL TEXTO(   1 ,ITEX5,1,28,32,2,3,IER)
      CALL TEXTO(   2 ,ITEX5,1,28,32,2,3,IER)
      CALL TEXTO(   1 ,ITEX6,3,32,28,2,3,IER)
      CALL TEXTO(   2 ,ITEX6,3,32,28,2,3,IER)
      CALL TEXTR(1,ITEX8,5,40,25,2,3,IER)
      READ(5,205) ALPHAO
      READ(5,201) NLSHIP
      DO 1 I=1,NLSHIP
      READ(5,202)  XSSHP(I),YSSHP(I),ZSSHP(I),XESHP(I),YESHP(I),
     1ZESHP(I)
    1 CONTINUE
      READ(5,201) NAC
      DO 2 I=1,NAC
      READ(5,208) XAC(I),YAC(I),ZAC(I),MD(I)
    2 CONTINUE
      READ(5,201) NLAND
      DO 35 I=1,NLAND
      READ(5,202) XSLND(I),YSLND(I),ZSLND(I),XELND(I),YELND(I),
     1ZELND(I)
   35 CONTINUE
      READ(5,203)XO,YO,ZO,S,F
      READ(5,205) YVPLSO
      READ(5,102) XINIT,ZINIT
      READ(5,203) WIRE1,WIRE2,WIRE3,WIRE4,STERN
      NAC1=NAC+1
      ALPHAO=ALPHAO/57.296
      NLC=NLAND+1
      NLC1=2*NLC
      NLC2=2*NLC+1
      ONEDEG=1.0/57.296
      TWODEG=2.0/57.296
      ONEHAF=1.5/57.296
      UZERO=UO
      CNST3=0.1 /ONEHAF
      CNST4=0.5/75.0
      CNST5=XINIT+STERN
C BUILD MOVE DRAW ARRAY FOR RUNWAY GENERATION
      DO 36 I=1,NLC
      MDC(2*I-1)=0
      MDC(2*I)=1
   36 CONTINUE
```

```
MEATBALL ARRAY BUILCING
      AMBX(1)=AMBX(4)=AMBX(5)=-1.08
      AMBX(2)=AMBX(3)=-1.12
      IMBMD(1)=0
      DO 96 I=2,5
      IMBMD(I)=1
   96 CONTINUE
BUILD FIXED GRAPHICS FOR VIEW FROM THE COCKPIT DISPLAY
   98 IF (MOD(ITDIR1(71),8).EQ.0) GO TO 98
  481 CONTINUE
      CALL DTINIT(2,ITDIR2,20,IER)
      CALL DTINIT(1,ITDIR1,20,IER)
      ENCODE(36,217,ITEX9)
      ENCODE(32,218,ITEX10)
      ENCODE(32,219,ITEX11)
      ENCODE(12,220,ITEX12)
      ENCODE(32,221,ITEX13)
      ENCODE(36,222,ITEX14)
      ENCODE(32,223,ITEX15)
      ENCODE(36,224,ITEX16)
      ENCODE(32,225,ITEX17)
      ENCODE(12,226,ITEX18)
      ENCODE(36,227,ITEX19)
************************************************************************
*                                                                      *
*           DISFLAY  MENU                                              *
*                                                                      *
************************************************************************
      ENCODE(32,228,ITEX20)
      ENCODE(32,229,ITEX21)
      ENCODE(36,230,ITEX22)
      ENCODE( 8,231,ITEX23)
      CALL TEXTO(2,ITEX9 ,9,10,10,2,3,IER)
      CALL TEXTO(2,ITEX10,8,14,10,2,3,IER)
      CALL TEXTO(2,ITEX11,8,18,10,2,3,IER)
      CALL TEXTO(2,ITEX12,3,22,10,2,3,IER)
      CALL TEXTO(1,ITEX13,8, 2,10,2,3,IER)
      CALL TEXTO(1,ITEX14,9, 5,10,2,3,IER)
      CALL TEXTO(1,ITEX15,8,10,10,2,3,IER)
      CALL TEXTO(1,ITEX16,9,13,10,2,3,IER)
      CALL TEXTO(1,ITEX17,8,16,10,2,3,IER)
      CALL TEXTO(1,ITEX18,3,19,10,2,3,IER)
      CALL TEXTO(1,ITEX19,9,24,10,2,3,IER)
      CALL TEXTO(1,ITEX20,8,27,10,2,3,IER)
      CALL TEXTO(1,ITEX21,8,32,10,2,3,IER)
      CALL TEXTO(1,ITEX22,9,35,10,2,3,IER)
      CALL TEXTO(1,ITEX23,2,38,10,2,3,IER)
      CALL TEXTR(   1  ,ITEX7,1,40,48,1,1,IER)
```

```
496 IF (MOD(ITCIR1(12),8).EQ.0) GO TO 496
*****************************************************************
*                                                               *
*                     DISPLAY PILOTS WINDOW                     *
*                                                               *
*****************************************************************
      ISQ(1)=IHEAD(0,10)
      ISQ(2)=IPACK(+1.0,+1.0,0)
      ISQ(3)=IPACK(+1.0,-1.0,1)
      ISQ(4)=IPACK(-1.0,-1.0,1)
      ISQ(5)=IPACK(-1.0,+1.0,1)
      ISQ(6)=IPACK(+1.0,+1.0,1)
      ISQ(7)=IPACK(-1.0,0.0,0)
      ISQ(8)=IPACK(-1.0,0.04,1)
      ISQ(9)=IPACK(-0.8,0.04,1)
      ISQ(10)=IPACK(-0.8,0.0,1)
      ISQ(11)=IPACK(-1.2,0.0,0)
      ISQ(12)=IPACK(-1.4,0.0,1)
      ISQ(13)=IPACK(-1.4,0.04,1)
      ISQ(14)=IPACK(-1.2,0.04,1)
      ISQ(15)=IPACK(-1.2,0.0,1)
      ISQ(16)=IPACK(-1.0,0.0,0)
      ISQ(17)=IPACK(-0.8,0.0,1)
      ISQ(18)=IPACK(1.0,0.0,0)
      ISQ(19)=IPACK(1.2,0.0,1)
      ISQ(20)=IPACK(-1.1,(10.0/18.5),0)
      ISQ(21)=IPACK(-1.0,(10.0/18.5),1)
      ISQ(22)=IPACK(1.0,(10.0/18.5),0)
      ISQ(23)=IPACK(1.1,(10.0/18.5),1)
      ISQ(24)=IPACK(-1.1,(-10.0/18.5),0)
      ISQ(25)=IPACK(-1.0,(-10.0/18.5),1)
      ISQ(26)=IPACK(1.0,(-10.0/18.5),0)
      ISQ(27)=IPACK(1.1,(-10.0/18.5),1)
      IFAST(1)=IHEAD(0,10)
      IFAST(2)=IPACK(1.04,0.8,0)
      IFAST(3)=IPACK(1.12,0.88,1)
      IFAST(4)=IPACK(1.2,0.8,1)
      ICN(1)=IHEAD(0,10)
      ICN(2)=IPACK(1.08,0.96,0)
      ICN(3)=IPACK(1.08,1.04,1)
      ICN(4)=IPACK(1.16,1.04,1)
      ICN(5)=IPACK(1.16,0.96,1)
      ICN(6)=IPACK(1.08,0.96,1)
      ISLO(1)=IHEAD(0,10)
      ISLO(2)=IPACK(1.04,1.2,0)
      ISLO(3)=IPACK(1.12,1.12,1)
      ISLO(4)=IPACK(1.2,1.2,1)
      IBCGUS(1)=IHEAD(0,10)
```

```
      CC 474 I=2,50
      IBCGUS(I)=IPACK(1.0,1.0,0)
  474 CCNTINUE
      CALL DTINIT(1,ITDIR1,20,IER)
      CALL DTINIT(2,ITDIR2,20,IER)
      CALL GRAPHC(2,ISQ,27,1,IER)
TRANSLATICN AND ROTATICN CF CBSERVER
      Y=YVPLSC/57.296
      CYAW=CCS(Y)
      SYAW=SIN(Y)
      CC 3 I=1,NLSHIP
      XSSHIP(I)=XSSHP(I)-XO
      XESHIP(I)=XESHP(I)-XO
      YSSHIP(I)=YSSHP(I)-YO
      YESHIP(I)=YESHP(I)-YO
      ZSSHIP(I)=ZSSHP(I)-ZO
      ZESHIP(I)=ZESHP(I)-ZO
      XS1(I)=XSSHIP(I)*CYAW+YSSHIP(I)*SYAW
      XE1(I)=XESHIP(I)*CYAW+YESHIP(I)*SYAW
      YS1(I)=-XSSHIP(I)*SYAW+YSSHIP(I)*CYAW
      YE1(I)=-XESHIP(I)*SYAW+YESHIP(I)*CYAW
      ZS1(I)=ZSSHIP(I)
      ZE1(I)=ZESHIP(I)
    3 CCNTINUE
      NL1=NLSHIP+1
      NL2=NLSHIP+2
THE HCRIZON
      XS1(NL1)=XE1(NL1)=XS1(NL2)=XE1(NL2)=-1000CCC.
      ZS1(NL1)=ZE1(NL1)=ZS1(NL2)=ZE1(NL2)=0.0
      YS1(NL1)=-10C0000.
      YE1(NL1)=YS1(NL2)=0.0
      YE1(NL2)=1CC0000.
VIEWING PLANE CHCP
      CC 4 I=1,NL2
      IF (XS1(I).GT.0.0) GO TO 800
      IF (XE1(I).LE.0.0) GO TO 4
      YE1(I)=YS1(I)-XS1(I)*(YE1(I)-YS1(I))/(XE1(I)-XS1(I))
      ZE1(I)=ZS1(I)-XS1(I)*(ZE1(I)-ZS1(I))/(XE1(I)-XS1(I))
      XE1(I)=0.0
      GC TO 4
  8CC IF (XE1(I).GT.0.0) GO TC 805
      YS1(I)=YE1(I)-XE1(I)*(YS1(I)-YE1(I))/(XS1(I)-XE1(I))
      ZS1(I)=ZE1(I)-XE1(I)*(ZS1(I)-ZE1(I))/(XS1(I)-XE1(I))
      XS1(I)=C.0
      GC TO 4
  805 XS1(I)=XE1(I)=0.0
      YS1(I)=YE1(I)=ZS1(I)=ZE1(I)=-1.4
    4 CCNTINUE
```

```
PERSPECTIVE
      DC 5 I=1,NL2
      W2    =-S*XS1(I)+F*S
      XS2(I)=F*XS1(I)/W2
      YS2(I)=F*YS1(I)/W2
      ZS2(I)=F*ZS1(I)/W2
      W2    =-S*XE1(I)+F*S
      XE2(I)=F*XE1(I)/W2
      YE2(I)=F*YE1(I)/W2
      ZE2(I)=F*ZE1(I)/W2
    5 CCNTINUE
THE WINDCW
      DC 6 I=1,NL2
      IY1=YS2(I)
      IY2=YE2(I)
      IZ1=ZS2(I)
      IZ2=ZE2(I)
      IF (IY1.GE.1.AND.IY2.GE.1) GO TC  61
      IF (IY1.LE.-1.AND.IY2.LE.-1) GO TC  61
      IF (IZ1.GE.1.AND.IZ2.GE.1) GO TO  61
      IF (IZ1.LE.-1.AND.IZ2.LE.-1) GO TC  61
      IAGN=1
      Y=YS2(I)
      Z=ZS2(I)
      IY=IY1
      IZ=IZ1
      SLCPE=(ZS2(I)-ZE2(I))/(YS2(I)-YE2(I))
    7 IF(IY) 11, 12, 13
   11 VINSCT=Z+SLOPE*(-1.4-Y)
   21 IF (ABS(VINSCT).LE.1.4) GO TO  26
   23 IF (IZ)24,61,25
   26 YTEMP=-1.4
      GC TO  47
   24 HINSCT=Y+(-1.4-Z)/SLOPE
      ZTEMP=-1.4
   22 IF (ABS(HINSCT).GT.1.4) GO TO  61
      YTEMP=HINSCT
      GC TO  50
   25 HINSCT=Y+(1.4-Z)/SLOPE
      ZTEMP=1.4
      GC TO  22
   12 IF(IZ)24,32,25
   32 YTEMP=Y
      ZTEMP=Z
      GC TO  50
   13 VINSCT=Z+SLOPE*(1.4-Y)
      IF (ABS(V.INSCT).LE.1.4) GO TO  46
      GC TO  23
```

```
   46 YTEMP=1.4
   47 ZTEMP=VINSCT
   50 GO TO ( 51, 60),IAGN
   51 Y=YE2(I)
      Z=ZE2(I)
      IY=IY2
      IZ=IZ2
      IAGN=2
      YS2(I)=YTEMP
      ZS2(I)=ZTEMP
      GO TO 7
   60 CONTINUE
      YE2(I)=YTEMP
      ZE2(I)=ZTEMP
      GO TO 6
   61 YE2(I)=-1.4
      YS2(I)=-1.4
      ZE2(I)=-1.4
      ZS2(I)=-1.4
    6 CONTINUE
DIVIDE THE HORIZON INTO 4 PARTS
      NL3=NLSHIP+3
      NL4=NLSHIP+4
      ZS2(NL3)=ZE2(NL3)=ZS2(NL4)=ZE2(NL4)=0.0
      TEMP1=YS2(NL1)/2.0
      TEMP2=YE2(NL2)/2.0
      YE2(NL4)=YE2(NL2)
      YE2(NL1)=YS2(NL2)=TEMP1
      YE2(NL2)=YS2(NL3)=0.0
      YE2(NL3)=YS2(NL4)=TEMP2
DISPLAY GRAPHICS
      IPIC(1)=IHEAD(0,10)
      DO 8 I=1,NL4
      IPIC(2*I)=IPACK(YS2(I),ZS2(I),0)
      IPIC(2*I+1)=IPACK(YE2(I),ZE2(I),1)
    8 CONTINUE
      NB=2*NL4+1
      CALL GRAPHC(   1,IPIC,NB,1,IER)
*                                                                      *
************************************************************************
*                                                                      *
*                     DISPLAY CARRIER DECK TO LSC                      *
*                                                                      *
************************************************************************
AIRCRAFT REGENERATING SCHEME
      CNST1=ALPHA0+THO
      CHECK=-XINIT+XO
      FLG=0.0
```

```
      CALL RESET(500)
      CALL COMPUTE
      LL=0
      DO 400 L=1,400
      LL=LL+1
      DC 906 L1=1,10
BUILC THE HORIZON FOR THE CARRIER
      XS2(NLC)=XE2(NLC)=-1000000.
      ZS2(NLC)=ZE2(NLC)=-ZINIT
      YS2(NLC)=-1C00000.
      YE2(NLC)=1000000.
      CALL ADK(0,X,1,Y,2,Z,3,W,4,THETA,5,PSI,6,PHI)
      X=X*SCX
      Y=Y*SCY
      Z=Z*SCZ
      W=W*SCW
      THETA=THETA*SCTH
      PSI=PSI*SCSI
      PHI=PHI*SCPH
      CNST2=CNST1+THETA
      XKEEP(LL)=-XINIT-X
      YKEEP(LL)=Y
      ZKEEP(LL)=ZINIT-Z
      IF (FLG.GT.1.0) GO TO 480
OBJECT AXIS (A/C) ROTATION
      DC 15 I=1,NAC
      XAC1(I)=XAC(I)+PSI*YAC(I)-CNST2*ZAC(I)
      YAC1(I)=-XAC(I)*PSI+YAC(I)-PHI*ZAC(I)
      ZAC1(I)=CNST2*XAC(I)+PHI*YAC(I)+ZAC(I)
   15 CCNTINUE
TRANSLATICN
      XT=X+XINIT-XO
      YT=-(Y+YO)
      ZT=ZINIT-Z-ZO
      DC 16 I=1,NAC
      XAC2(I)=XAC1(I)+XT
      YAC2(I)=YAC1(I)+YT
      ZAC2(I)=ZAC1(I)+ZT
   16 CCNTINUE
VIEWING PLANE (LSO) ROTATION
      DO 17 I=1,NAC
      XAC3(I)=XAC2(I)*CYAW+YAC2(I)*SYAW
      YAC3(I)=-XAC2(I)*SYAW+YAC2(I)*CYAW
   17 CCNTINUE
PRCSPECTIVE
      DC 18 I=1,NAC
      W2=1.0/(-S*XAC3(I)+F*S)
      YAC4(I)=F*YAC3(I)*W2
```

```
      ZAC4(I)=F*ZAC2(I)*W2
   18 CONTINUE
      IF (ABS(YAC4(1)).GT.1.8) GO TO 480
      IF (ABS(ZAC4(1)).GT.1.8) GO TO 480
DISPLAY GRAPHICS
      CALL RTCI(NAC,YAC4,ZAC4,MD,IAC,0,10)
      CALL GRAPHO(    1,IAC,NAC1,2,IER)
      GO TO 479
  480 CONTINUE
      CALL GRAPHO(1,IBOGUS,49,2,IER)
      FLG=2.0
  479 CONTINUE
      ZINIT1=ZINIT+30.0
      IF (Z.GT.ZINIT1) GO TO 482
      IF ((X.GT.-CNST5).AND.(Z.GT.ZINIT)) GO TO 483
      IF (X.GT.(SCX-100.0)) GO TO 484
      ZSAVE=Z
TRANSLATION OF THE CARRIER
      XT1=XINIT+X
      YT1=-Y
      ZT1=-ZINIT+Z
      DO 37 I=1,NLAND
      XS2(I)=XSLND(I)+XT1
      XE2(I)=XELND(I)+XT1
      YS2(I)=YSLND(I)+YT1
      YE2(I)=YELND(I)+YT1
      ZS2(I)=ZSLND(I)+ZT1
      ZE2(I)=ZELND(I)+ZT1
   37 CONTINUE
ROTATION (VP) OF THE CARRIER
      DO 38 I=1,NLC
      XS1(I)=XS2(I)-YS2(I)*PSI-ZS2(I)*CNST2
      XE1(I)=XE2(I)-YE2(I)*PSI-ZE2(I)*CNST2
      YS1(I)=+XS2(I)*PSI+YS2(I)-ZS2(I)*PHI
      YE1(I)=+XE2(I)*PSI+YE2(I)-ZE2(I)*PHI
      ZS1(I)=+XS2(I)*CNST2+YS2(I)*PHI+ZS2(I)
      ZE1(I)=+XE2(I)*CNST2+YE2(I)*PHI+ZE2(I)
   38 CONTINUE
VIEWING PLANE CHOP
      DO 39 I=1,NLC
      IF (XS1(I).GT.0.0) GO TO 801
      IF (XE1(I).LE.0.0) GO TO 39
      YE1(I)=YS1(I)-XS1(I)*(YE1(I)-YS1(I))/(XE1(I)-XS1(I))
      ZE1(I)=ZS1(I)-XS1(I)*(ZE1(I)-ZS1(I))/(XE1(I)-XS1(I))
      XE1(I)=0.0
      GO TO 39
  801 IF (XE1(I).GT.0.0) GO TO 802
      YS1(I)=YE1(I)-XE1(I)*(YS1(I)-YE1(I))/(XS1(I)-XE1(I))
```

```
      ZS1(I)=ZE1(I)-XE1(I)*(ZS1(I)-ZE1(I))/(XS1(I)-XE1(I))
      XS1(I)=C.0
      GC TO 39
  802 XS1(I)=XE1(I)=0.0
      YS1(I)=YE1(I)=ZS1(I)=ZE1(I)=-1.
   39 CCNTINUE
PERSPECTIVE
      DC 40 I=1,NLC
      W2    =-S*XS1(I)+F*S
      XS2(I)=F*XS1(I)/W2
      YS2(I)=F*YS1(I)/W2
      ZS2(I)=F*ZS1(I)/W2
      W2    =-S*XE1(I)+F*S
      XE2(I)=F*XE1(I)/W2
      YE2(I)=F*YE1(I)/W2
      ZE2(I)=F*ZE1(I)/W2
   4C CCNTINUE
THE WINDCW
      DC 41 I=1,NLC
      IY1=YS2(I)
      IY2=YE2(I)
      IZ1=ZS2(I)
      IZ2=ZE2(I)
      IF (IY1.GE.1.AND.IY2.GE.1) GO TO 461
      IF (IY1.LE.-1.AND.IY2.LE.-1) GO TC 461
      IF (IZ1.GE.1.AND.IZ2.GE.1) GO TO 461
      IF (IZ1.LE.-1.AND.IZ2.LE.-1) GO TC 461
      IAGN=1
      Y=YS2(I)
      Z=ZS2(I)
      IY=IY1
      IZ=IZ1
      SLCPE=(ZS2(I)-ZE2(I))/(YS2(I)-YE2(I))
  401 IF(IY)411,412,413
  411 VINSCT=Z+SLCPE*(-1.0-Y)
  421 IF (ABS(VINSCT).LE.1.) GO TO 426
  423 IF (IZ)424,461,425
  426 YTEMP=-1.
      GC TO 447
  424 HINSCT=Y+(-1.-Z)/SLOPE
      ZTEMP=-1.0
  422 IF (ABS(HINSCT).GT.1.) GO TO 461
      YTEMP=HINSCT
      GC TO 450
  425 HINSCT=Y+(1.-Z)/SLOPE
      ZTEMP=1.
      GC TO 422
  412 IF(IZ)424,432,425
```

```
  432 YTEMP=Y
      ZTEMP=Z
      GO TO 450
  413 VINSCT=Z+SLOPE*(1.-Y)
      IF (ABS(VINSCT).LE.1.) GO TO 446
      GO TO 423
  446 YTEMP=1.
  447 ZTEMP=VINSCT
  450 GO TO (451,460),IAGN
  451 Y=YE2(I)
      Z=ZE2(I)
      IY=IY2
      IZ=IZ2
      IAGN=2
      YS2(I)=YTEMP
      ZS2(I)=ZTEMP
      GO TO 401
  460 CONTINUE
      YE2(I)=YTEMP
      ZE2(I)=ZTEMP
      GO TO 41
  461 YE2(I)=-1.
      YS2(I)=-1.
      ZE2(I)=-1.
      ZS2(I)=-1.
   41 CONTINUE
DISPLAY GRAPHICS
      DO 42 I=1,NLC
      J=2*I
      K=J-1
      YS1(K)=YS2(I)
      YS1(J)=YE2(I)
      ZS1(K)=ZS2(I)
      ZS1(J)=ZE2(I)
   42 CONTINUE
ANGLE OF ATTACK INDICATOR
      AOA=W/UO
      IF (AOA.GT.TWODEG) GO TO 93
      IF (AOA.LT.-TWODEG) GO TO 92
      IF (AOA.GT.ONEDEG) GO TO 91
      IF (AOA.LT.-ONEDEG) GO TO 90
ON AIRSPEED
      CALL GRAPHC(    2,ICN,6,2,IER)
      GO TO 99
   90 DO 94 I=2,4
      J=I+5
      ICN(J)=IFAST(I)
   94 CONTINUE
```

```
      CALL GRAPHO(   2,ICN,9,2,IER)
      GO TO 99
   91 DO 95 I=2,4
      J=I+5
      ICN(J)=ISLO(I)
   95 CONTINUE
      CALL GRAPHO(   2,ICN,9,2,IER)
      GO TO 99
   92 CALL GRAPHO(   2,IFAST,4,2,IER)
      GO TO 99
   93 CALL GRAPHO(   2,ISLO,4,2,IER)
   99 CONTINUE
MEATBALL DISPLAY
      Z=ZSAVE
      ANGL=(ZINIT-Z)/(XINIT+X)
      DIFF=THC-ANGL
      YMB=DIFF*CNST3
      IF (DIFF.LT.-ONEHAF) GO TO 97
      IF (DIFF.GT.ONEHAF) GO TO 97
      NLC5=NLC1+5
      NLC6=NLC1+6
      DO 901 I=1,5
      YS1(NLC1+I)=AMBX(I)
      MDC(NLC1+I)=IMBMD(I)
  901 CONTINUE
      ZS1(NLC1+1)=ZS1(NLC1+2)=ZS1(NLC5)=YMB
      ZS1(NLC1+4)=ZS1(NLC1+3)=YMB+0.04
      CALL RTOI(NLC5,YS1,ZS1,MDC,ILAND,0,10)
      CALL GRAPHO(2,ILAND,NLC6,3,IER)
      GO TO 906
   97 CONTINUE
      DO 902 I=NLC2,NLC5
      MDC(I)=0
  902 CONTINUE
      CALL RTOI(NLC5,YS1,ZS1,MDC,ILAND,0,10)
      CALL GRAPHO(2,ILAND,NLC6,3,IER)
  906 CONTINUE
  400 CONTINUE
  900 CONTINUE
*****************************************************************************
CRASH
  482 CONTINUE
      CALL HOLD
  475 CALL DGINIT(1,IGDIR1, 9,IER)
      CALL DGINIT(2,IGDIR2, 9,IER)
      ENCODE(8,232,ITEX24)
      CALL TEXTO(1,ITEX24,2,20,44,3,3,IER)
      CALL TEXTO(2,ITEX24,2,20,44,3,3,IER)
```

```
      CALL TEXTR(1,ITEX8,5,40,25,2,3,IER)
  478 IF (MOD(ITCIR1(2),8).EQ.0) GO TC 478
      GC TO 485
LANCING CATA
  483 ENCCDE(24,233,ITEX25)
      CALL HOLD
      IF (ABS(Y).GT.50.0) GO TO 475
      CALL DGINIT(1,IGDIR1, 9,IER)
      CALL DGINIT(2,IGDIR2, 9,IER)
RAMF STRIKE
      IF (Z.LT.ZINIT+ 5.0) GC TO 634
      ENCODE(12,261,ITEX52)
      CALL TEXTO(1,ITEX52,3,20,20,3,3,IER)
      CALL TEXTR(1,ITEX8,5,40,25,2,3,IER)
  635 IF (MOD(ITCIR1( 2),8).EQ.0) GO TC 635
      GC TO 485
  634 CCNTINUE
EXCESSIVE RATE OF DECENT
      CALL ADK(7,ZVEL1)
      ZVEL1=ZVEL1*SCV
      ZVEL2=UO*STHO
      ZVEL=(ZVEL1+ZVEL2)*60.0
      IF (ZVEL.LT.720.0) GO TO 636
      ENCCDE(28,262,ITEX53)
      ENCCDE(40,263,ITEX54) ZVEL
      CALL TEXTO(1,ITEX53,7,15,5,2,3,IER)
      CALL TEXTO(2,ITEX53,7,15,5,2,3,IER)
      CALL TEXTO(1,ITEX54,10,25,1,2,3,IER)
      CALL TEXTO(2,ITEX54,10,25,1,2,3,IER)
      CALL TEXTR(1,ITEX8,5,40,25,2,3,IER)
  637 IF (MOD(ITDIR1(3),8).EQ.0) GO TC 637
      GC TO 485
  636 CCNTINUE
      ENCODE(16,257,ITEX49)
      CALL TEXTO(1,ITEX25,6,10,30,2,3,IER)
      CALL TEXTO(2,ITEX25,6,10,30,2,3,IER)
      IF (XKEEP(LL).GT.WIRE1) GO TO 81
      IF (XKEEP(LL).GT.WIRE2) GO TO 82
      IF (XKEEP(LL).GT.WIRE3) GO TC 83
      IF (XKEEP(LL).GT.WIRE4) GO TO 84
      CALL TEXTO(1,ITEX49,4,25,10,2,3,IER)
      CALL TEXTO(2,ITEX49,4,25,10,2,3,IER)
      CALL TEXTR(1,ITEX8,5,40,25,2,3,IER)
  620 IF (MOD(ITCIR1( 3),8).EQ.0) GO TC 620
      GC TO 86
   81 IWIRE=1
      GC TO 85
   82 IWIRE=2
```

```
      GO TO 85
   83 IWIRE=3
      GO TO 85
   84 IWIRE=4
   85 ENCODE(16,258,ITEX50) IWIRE
      CALL TEXTO(1,ITEX50,4,17,10,2,3,IER)
      CALL TEXTO(2,ITEX50,4,17,10,2,3,IER)
      ENCODE(40,264,ITEX55) ZVEL
      CALL TEXTO(1,ITEX55,10,22,10,2,3,IER)
      CALL TEXTO(2,ITEX55,10,22,10,2,3,IER)
      IF (YKEEP(LL).LT.0.0) GO TO 87
      ENCODE(40,259,ITEX51) YKEEP(LL)
      GO TO 88
   87 ENCODE(40,260,ITEX51) YKEEP(LL)
   88 CALL TEXTO(1,ITEX51,10,27,10,2,3,IER)
      CALL TEXTO(2,ITEX51,10,27,10,2,3,IER)
      CALL TEXTR(1,ITEX8,5,40,25,2,3,IER)
  497 IF (MOD(ITDIR1( 5),8).EQ.0) GO TO 497
   86 CONTINUE
      GO TO 485
NO LANDING
  484 ENCODE(16,234,ITEX26)
      CALL HOLD
      CALL DGINIT(1,IGDIR1, 9,IER)
      CALL DGINIT(2,IGDIR2, 9,IER)
      CALL TEXTO(1,ITEX26,4,20,15,2,3,IER)
      CALL TEXTO(2,ITEX26,4,20,15,2,3,IER)
      CALL TEXTR(1,ITEX8,5,40,25,2,3,IER)
  498 IF (MOD(ITDIR1( 2),8).EQ.0) GO TO 498
  485 CONTINUE
MENUE FOR RESTART
      CALL DTINIT(1,ITDIR1,20,IER)
      ENCODE(36,235,ITEX27)
      ENCODE(36,236,ITEX28)
      ENCODE(24,237,ITEX29)
      ENCODE(36,238,ITEX30)
      ENCODE(36,239,ITEX31)
      ENCODE(32,240,ITEX32)
      CALL TEXTO(1,ITEX27,9, 5,10,2,3,IER)
      CALL TEXTO(1,ITEX30,9,15,10,2,3,IER)
      CALL TEXTO(1,ITEX31,9,18,10,2,3,IER)
      CALL TEXTO(1,ITEX32,8,21,10,2,3,IER)
      CALL TEXTO(1,ITEX28,9,31,10,2,3,IER)
      CALL TEXTO(1,ITEX29,6,33,10,2,3,IER)
      CALL TEXTR(1,ITEX7,1,40,48,1,3,IER)
  499 IF (MOD(ITDIR1( 7),8).EQ.0) GO TO 499
      CALL TEXTI(1,I1,1,40,48,IER)
      DECODE(4,207,I1) I1
```

```
      IF (I1.EQ.1) GO TO 920
      IF (I1.EQ.2) GO TO 495
      GO TO 481
  920 CONTINUE
DISPLAY RESULTS
      CALL DTINIT(1,ITDIR1,20,IER)
      CALL DGINIT(1,IGDIR1, 9,IER)
      I5=0
  910 I5=I5+1
      IF (XKEEP(I5).GE.2000.0) GO TO 910
      J=0
      DO 911 I=I5,LL
      IF (XKEEP(I).LT.0.0) GO TO 921
      IF (ZKEEP(I).LT.0.0) GO TO 921
      J=J+1
      XKEEP(J)=(XKEEP(I)-1000.0)/1000.0
      YKEEP(J)=YKEEP(I)/100.0
      ZKEEP(J)=(ZKEEP(I)-200.0)/200.0
  911 CONTINUE
  921 IPLOT1(1)=IHEAD(0,10)
      IPLOT1(2)=IPACK(-1.0,1.0,0)
      IPLOT1(3)=IPACK(-1.0,-1.0,1)
      IPLOT1(4)=IPACK(1.0,-1.0,1)
      IPLOT1(5)=IPACK(-1.0,-0.5,0)
      IPLOT1(6)=IPACK(-1.05,-0.5,1)
      IPLOT1(7)=IPACK(-1.0,0.0,0)
      IPLOT1(8)=IPACK(-1.05,0.0,1)
      IPLOT1(9)=IPACK(-1.0,0.5,0)
      IPLOT1(10)=IPACK(-1.05,0.5,1)
      IPLOT1(11)=IPACK(-0.5,-1.0,0)
      IPLOT1(12)=IPACK(-0.5,-1.05,1)
      IPLOT1(13)=IPACK(0.0,-1.0,0)
      IPLOT1(14)=IPACK(0.0,-1.05,1)
      IPLOT1(15)=IPACK(0.5,-1.0,0)
      IPLOT1(16)=IPACK(0.5,-1.05,1)
      IPLOT1(17)=IPACK(-1.0,-1.0,0)
      YTEMP=-(ZINIT/XINIT)*2000.0
      IPLOT1(18)=IPACK(1.0,((YTEMP-200.0)/200.0),1)
      STERNG=(STERN-WIRE2-1000.0)/1000.0
      IPLOT1(19)=IPACK(STERNG,-1.0,0)
      IPLOT1(20)=IPACK(STERNG,-0.95,1)
      IPLOT2(1)=IHEAD(1,10)
      IPLOT2(2)=IPACK(XKEEP(1),ZKEEP(1),0)
      DO 912 I=1,J
      J1=I+2
      IPLOT2(J1)=IPACK(XKEEP(I),ZKEEP(I),1)
  912 CONTINUE
      J2=J+2
```

```
      CALL GRAPHO(1,IPLOT1,20,1,IER)
      CALL GRAPHO(1,IPLOT2,J2,2,IER)
      ENCODE(4,241,ITEX33)
      ENCODE(4,242,ITEX34)
      ENCODE(4,243,ITEX35)
      ENCODE(4,244,ITEX36)
      ENCODE(4,245,ITEX37)
      ENCODE(4,246,ITEX38)
      ENCODE(4,247,ITEX39)
      ENCODE(4,248,ITEX40)
      ENCODE(4,249,ITEX41)
      ENCODE(4,250,ITEX42)
      ENCODE(4,251,ITEX43)
      ENCODE(20,252,ITEX44)
      ENCODE(16,253,ITEX45)
      ENCODE(36,254,ITEX46)
      ENCODE(16,255,ITEX47)
      ENCODE(16,256,ITEX48)
      ENCODE(8,265,ITEX56)
      CALL TEXTO(1,ITEX36,1,29, 3,1,3,IER)
      CALL TEXTO(1,ITEX37,1,21, 3,1,3,IER)
      CALL TEXTO(1,ITEX38,1,13, 3,1,3,IER)
      CALL TEXTO(1,ITEX39,1, 5, 3,1,3,IER)
      CALL TEXTO(1,ITEX40,1,39,27,1,3,IER)
      CALL TEXTO(1,ITEX41,1,39,47,1,3,IER)
      CALL TEXTO(1,ITEX42,1,39,67,1,3,IER)
      CALL TEXTO(1,ITEX43,1,39,87,1,3,IER)
      CALL TEXTO(1,ITEX44,5, 4,20,2,3,IER)
      CALL TEXTO(1,ITEX46,9, 7,20,1,3,IER)
      CALL TEXTO(1,ITEX56,2,37,19,1,3,IER)
      CALL TEXTR(1,ITEX8,5,40,48,1,3,IER)
  916 IF (MOD(ITDIR1(12),8).EQ.0) GO TO 916
SECOND GRAPH
      CALL DTINIT(1,ITDIR1,20,IER)
      CALL DGINIT(1,IGDIR1, 9,IER)
      IPLOT1(17)=IPACK(-1.0,0.0,0)
      IPLOT1(18)=IPACK(1.0,0.0,1)
      CALL GRAPHO(1,IPLOT1,18,1,IER)
      IPLOT3(1)=IHEAD(1,10)
      IPLOT3(2)=IPACK(XKEEP(1),YKEEP(1),0)
      DO 913 I=1,J
      J1=I+2
      IPLOT3(J1)=IPACK(XKEEP(I),YKEEP(I),1)
  913 CONTINUE
      CALL GRAPHO(1,IPLOT3,J2,2,IER)
      CALL TEXTO(1,ITEX33,1,13, 3,1,3,IER)
      CALL TEXTO(1,ITEX34,1,29, 3,1,3,IER)
      CALL TEXTO(1,ITEX35,1,38, 1,1,3,IER)
```

```
      CALL TEXTO(1,ITEX36,1, 5, 3,1,3,IER)
      CALL TEXTO(1,ITEX40,1,39,27,1,3,IER)
      CALL TEXTC(1,ITEX41,1,39,47,1,3,IER)
      CALL TEXTO(1,ITEX42,1,39,67,1,3,IER)
      CALL TEXTO(1,ITEX43,1,39,87,1,3,IER)
      CALL TEXTO(1,ITEX45,4, 4,20,2,3,IER)
      CALL TEXTO(1,ITEX46,9, 7,20,1,3,IER)
      CALL TEXTO(1,ITEX47,4,25,50,1,3,IER)
      CALL TEXTO(1,ITEX48,4,18,50,1,3,IER)
      CALL TEXTR(1,ITEX8,5,40,48,1,3,IER)
  917 IF (MOD(ITDIR1(13),8).EQ.0) GO TO 917
      CALL DTINIT(1,ITDIR1,20,IER)
      CALL DGINIT(1,IGDIR1, 9,IER)
      CALL TEXTC(1,ITEX27,9,15,10,2,3,IER)
      CALL TEXTO(1,ITEX28,9,25,10,2,3,IER)
      CALL TEXTO(1,ITEX29,6,28,10,2,3,IER)
      CALL TEXTR(1,ITEX7,1,40,48,1,3,IER)
  914 IF (MOD(ITDIR1( 4),8).EQ.0) GO TO 914
      CALL TEXTI(1,I1,1,40,48,IER)
      DECODE(4,207,I1) I1
      IF (I1.EQ.2) GO TO 495
      GO TO 481
  495 OUTPUT(101)'TYPE 1 TO READ NEW DATA','TYPE 0 TO STOP'
      READ(101,200)ITEST
      IF (ITEST)550,550,551
  550 STOP
      END
¬ASSIGN X1=MT1A
¬LOAD XR,MAP
¬SAVE
¬DATA
```

LIST OF REFERENCES

1. De Laura, R. D., Electrical Engineering Computer Laboratory, Naval Postgraduate School, 1972.

2. Etkin, Bernard, Dynamics of Flight, p. 94-133, Wiley, 1959.

3. Desens, R. B., Computer Processing for Display of Three-Dimensional Structures, Electrical Engineer Thesis, Naval Postgraduate School, 1969.

4. Kahrs, J. H., A Fixed-Base Variable-Stability Carrier Approach Landing Simulator, Master of Science Thesis, Naval Postgraduate School, 1972.

5. Office of Naval Research Aeronautics Code 461 Report 785-1, The Landing Signal Officer: A Problem Analysis, by Gail J. Borden, v. I, May 1969.

6. Office of Naval Research Aeronautics Code 461 Report 785-1, The Landing Signal Officer: A Problem Analysis, by Gail J. Borden, v. II, p. A33-A50, May 1969.

7. Naval Air Defense Command Report R0111-01-01/MF 9-01, Toward a Pilot Model for Carrier Approach, March 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chairman, Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Asst. Professor M. H. Redlin Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5.	Electrical Engineering Computer Laboratory Naval Postgraduate School Monterey, California 93940	1
6.	Lieutenant P. Gary Stueck 1701 Belburn Drive Belmont, California 94002	1
7.	C. J. Mazza Code 3015 Air Vehicle Technology Department Naval Air Development Center Warminster, Pennsylvania 18974	1

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

REPORT TITLE

LSO/Pilot Interaction Simulator

DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis; June 1973

AUTHOR(S) (First name, middle initial, last name)

Phillip Gary Stueck

REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1973	86	7

a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

0. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

This thesis investigates the feasibility of simulating the Landing Signal Officer (LSO)/pilot interaction during the approach to a landing on an aircraft carrier. A simulator was created which duplicated the LSO's operational environment through the use of computer-generated visual displays. The LSO and the pilot were placed in this simulated carrier approach environment by 1) displaying a representation of the landing area plus a "meatball" and angle of attack information to the pilot while 2) simultaneously displaying the aircraft's approach to the LSO.

Test results demonstrated the basic feasibility of simulating the LSO/pilot interaction and its application as a research tool in studying LSO models, wave-off techniques and landing techniques.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Landing Signal Officer						
Carrier Landing Simulation						
Simulation						

Thesis 145177
S85726 Stueck
c.1
LSO/pilot interaction
simulator.